

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.В. Коваль

(підпис)

” ” _____ 2019р.

ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 “Програмна інженерія”

на тему: “Система моделювання залежності виходу водню від умов попередньої обробки сировини”

Виконав: студент IV курсу, групи ТІ-51

Бондаренко Андрій Володимирович

(прізвище, ім'я, по батькові)

(підпис)

Керівник ст.викладач Бандурка О.І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ — 2019

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший, бакалаврський

Напрямок підготовки 6.050103 “Програмна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.В. Коваль

(підпис)

” ” _____ 2019р.

ЗАВДАННЯ

на дипломну роботу студенту

Бондаренку Андрію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи “Система моделювання залежності виходу водню від умов попередньої обробки сировини”

керівник роботи Бандурка Олена Іванівна, ст. викладач

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від 22.05.2019р. № 1325-с

2. Строк подання студентом роботи 10 червня 2019р.

3. Вихідні дані до роботи розроблена система написана мовою JavaScript

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити) спроектувати архітектуру системи моделювання залежності виходу водню від умов попередньої обробки сировини, розробити програмне забезпечення, розробити інтерфейс користувача.

5. Перелік ілюстративного матеріалу титульний аркуш, поставлені задачі, функції розроблюваної системи, архітектура системи, інтерфейс користувача, приклад роботи системи, висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання ”_14_” _____ вересня _____ 2018 __р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	12.02.19	
2	Розробка архітектури та загальної структури системи	23.04.19	
3.	Розробка структур окремих підсистем	30.04.19	
4.	Підготовка матеріалів	02.05.19	
5.	Програмна реалізація системи	10.05.19	
6.	Захист програмного продукту	18.05.19	
7.	Оформлення пояснювальної записки	01.06.19	
8.	Передзахист	19.06.19	

Студент _____
(підпис)

Бондаренко А.В.
(прізвище та ініціали,)

Керівник роботи _____
(підпис)

Бандурка О. І.
(прізвище та ініціали,)

АНОТАЦІЯ

Мета роботи — дослідження залежності методів попередньої обробки сировини та основних параметрів процесу видобутку водню від компонентного складу сировини та створення відповідного програмного продукту. Виконано реалізацію системи моделювання залежності виходу водню від умов попередньої обробки сировини. Програмний продукт являє собою застосунок, що визначає оптимальні параметри процесу утворення водню на основі введених користувачем даних про сировину.

Результати дослідження доповідалися на науковій конференції.

Дипломну роботу виконано на 54 аркушах, вона містить 4 додатків та перелік посилань на використані джерела з 20 найменувань. У роботі наведено 13 рисунків та 2 таблиці.

Ключові слова: водень, попередня обробка, пар, луг, Vue.js, Node.js, MongoDB.

ABSTRACT

The purpose of the work is to study the dependence of the methods of preliminary processing of raw materials and the main parameters of the process of hydrogen extraction from the component composition of raw materials and the creation of the corresponding software product. Implementation of the control system for the production of hydrogen from renewable cellulosic raw materials. The software product is an application that defines optimal hydrogen generation process based on user input of raw materials.

The results of the study were reported at a scientific conference.

Thesis is completed on 54 sheets, it contains 4 applications and a list of references to used sources of 20 titles. There are 13 drawings and 2 tables in the work.

Keywords: hydrogen, pre-processing, steam, meadow, Vue.js, Node.js, MongoDB.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 ЗАДАЧА ПОБУДОВИ СИСТЕМИ МОДЕЛЮВАННЯ ЗАЛЕЖНОСТІ ВИХОДУ ВОДНЮ ВІД УМОВИ ПОПЕРЕДНЬОЇ ОБРОБКИ СИРОВИНИ.....	9
2 ІМІТАЦІЙНА МОДЕЛЬ ПРОЦЕСУ ВИДОБУТКУ ВОДНЮ.....	11
2.1 Попередня обробка целюлозовмісної сировини.....	15
2.1.1 Попередня обробка сировини паром.....	16
2.1.2 Попередня обробка сировини лугом.....	18
2.2 Отримання водню з целюлозовмісної сировини у анаеробному процесі.....	20
2.3 Вплив параметрів процесу конверсії відходів на вихід водню.....	21
Висновки до розділу	22
3. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ.....	23
3.1 Вибір архітектури програмного комплексу.....	23
3.2 Опис архітектури клієнтського застосунку.....	25
3.3 Опис інструментів розробки.....	26
3.3.1 Мова програмування JavaScript.....	27
3.3.2 Фреймворк Vue.js	27
3.3.3 Серверна платформа Node.js.....	28
3.3.4 Фреймворк EXPRESS.....	30
3.3.5 База даних MongoDB	30
3.3.6 Мова розмітки HTML та метамова Sass.....	33
3.3.7 Формат передачі даних JSON.....	35
3.4 Обґрунтування вибору програмної реалізації.....	37
Висновки до розділу.....	39
4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ.....	40
4.1 Опис функціональності системи.....	40
4.2 Опис бази даних.....	41
4.3 Інтерфейс користувача.....	44

Висновки до розділу.....	45
5. РОБОТА КОРИСТУВАЧА В СИСТЕМІ.....	46
5.1 Форма авторизації користувача.....	46
5.2 Кабінет користувача.....	47
Висновки до розділу.....	51
ВИСНОВКИ.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК 1.....	55
ДОДАТОК 2.....	57
ДОДАТОК 3.....	68
ДОДАТОК 4.....	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

СКБД — система керування базами даних;

API (англ. Application Programming Interface) — прикладний програмний інтерфейс;

БД — база даних;

SQL (англ. Select query language) — декларативна мова програмування для взаємодії користувача з базами даних;

HTML (англ. Hypertext Markup Language) — стандартна мова розмітки для створення веб-сторінок і веб-додатків;

CSS (англ. Cascading Style Sheets) — спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних.

ВСТУП

Сучасний світ переживає водневу енергетичну революцію, яка поступово змінює основи енергетики, зумовлює нові тенденції розвитку паливно-енергетичного комплексу та чинить значний вплив на енергетичну політику країн світу. Водень знаходить своє застосування в багатьох сферах промисловості. Рідкий водень застосовується як ракетне паливо та як охолоджувач, оскільки має найвищу теплопровідність з усіх газів. Застосування водню має великі перспективи як універсальне екологічно чисте джерело енергії у паливних елементах та комбінованих енергетичних системах[1]. Водневе паливо використовують в авіації, для морського транспорту. Використання водню є перспективним вектором розвитку енергетичного комплексу країни.

Сучасні технології одержання водню є енергоємними та економічно не раціональними. Аналіз сучасного стану технологічних рішень одержання біоводню за використання відновлюваної сировини дозволяє констатувати, що основні проблеми виникають на етапі попередньої обробки целюлозовмісної сировини, оскільки існуючі технології не враховують особливостей компонентного складу сировини.

Метою дослідження є створення програмного застосунку що моделюватиме залежності виходу водню від умов попередньої обробки сировини. Система має вирішити проблему визначення залежності процесу продукування водню від компонентного складу відновлюваної сировини та методів її попередньої обробки і як результат, має встановити технологічні параметри процесу одержання біоводню.

1 ЗАДАЧА ПОБУДОВИ СИСТЕМИ МОДЕЛЮВАННЯ ЗАЛЕЖНОСТІ ВИХОДУ ВОДНЮ ВІД УМОВІ ПОПЕРЕДНЬОЇ ОБРОБКИ СИРОВИНИ

Світовий рівень промислового добування водню в наш час досягає майже 60 мільйонів тонн на рік. Ферментативне виробництво водню — це ферментативне перетворення органічного субстрату в біоводень, що здійснюється групою бактерій за допомогою мультиферментативних систем в три кроки, аналогічно до анаеробного перетворення[2]. Темнова ферментація не потребує світлової енергії, тому можливе неперервне виробництво водню з органічних сполук — вдень і вночі. Фотоферментація відрізняється від темної ферментації тим, що вона протікає лише за наявності світла. Біоводень може вироблятися у біореакторі, що містить водорості. Наприкінці 1990-х років було виявлено, що якщо з водоростей вилучити сульфур, вони вироблятимуть водень замість кисню, як під час звичайного фотосинтезу.

В сучасній Україні існує ряд проблем у сфері біоенергетики. Станом на сьогодні не існує промислового біотехнологічного виробництва водню за використання відходів. У складі угруповання мікроорганізмів, що перетворюють органічні речовини, містяться як продуценти водню, так і його консументи (метаногенні та сульфатредукуючі мікроорганізми). Тому для запобігання поглинання водню і збільшення його виходу необхідно усунути розвиток мікроорганізмів – споживачів молекулярного водню[3].

Тому було запропоновано дослідити процес видобутку водню та основні методи попередньої обробки сировини для вирішення поставленої задачі шляхом встановлення найбільш раціональних параметрів процесу видобутку водню з відновлюваної сировини.

Як результат дослідження, потрібно створити систему, основними функціями якої є:

- визначення методу попередньої обробки на основі введених користувачем даних;
- встановлення технологічних параметрів стадії попередньої обробки сировини;
- встановлення технологічних параметрів для основної стадії процесу видобутку водню;
- прогнозування можливої кількості водню, отриманого в результаті технологічної обробки заданої кількості сировини;
- побудова графіків можливого виходу водню;
- опис процесу видобутку водню з встановленими параметрами на кожному етапі видобутку водню.

Система має прогнозувати можливу кількість водню, отриманого в результаті технологічної обробки заданої кількості відновлюваної целюлозовмісної сировини, аналізуючи склад сировини, її кількість, метод первинної обробки та інших параметрів процесу утворення водню.

Програмний застосунок повинен забезпечити раціональне використання економічних та природних ресурсів, прискорити процес виробництва та збільшити обсяги готового енергетичного продукту.

2 ІМІТАЦІЙНА МОДЕЛЬ ПРОЦЕСУ ВИДОБУТКУ ВОДНЮ

Пошук альтернативних поновлюваних джерел енергії вже давно розглядається як пріоритетний напрям розвитку промислового і наукового прогресу людства. Важливе місце в цій сфері займають технології так званої «зеленої енергетики», коли для виробництва палива або прямого перетворення в теплову енергію використовується накопичена в результаті фотосинтезу енергія, яка міститься в біомасі вищих рослин. Перспективним кінцевим продуктом «зеленої енергетики» є біогаз суміш газів з переважним вмістом водню, що виходять шляхом анаеробного зброджування довільного органічної сировини в спеціальних установках біореакторах[4]. Як правило, сировиною в таких установках служать різні відходи часткового виробництва, що містять органічну речовину. Найчастіше це побічні продукти тваринництва. У той же час прийнято вважати, що цілеспрямоване культивування рослин як джерел біомаси, яка використовується як сировина для біогазових реакторів в загальному і цілому, виявляється економічно малоефективно. По-перше, енергоємність зеленої біомаси навіть у спеціальних енергетичних культур набагато нижче, ніж, наприклад, у компосту. по-друге, набагато більшою в цьому випадку виявляється тривалість виробничого циклу переробки. Істотно більш вигідним в цьому сенсі виявляється вирощування олійних енергетичних культур і їх переробка в рідке біопаливо. Підхід, здатний реанімувати ідею виробництва біогазу з рослинної сировини в масштабі середнього рослинницької господарства, полягає в тому, що контур виробництва біогазу пропонується впровадити у виробничий цикл рослинницької господарства не замість традиційних сівозмін, а разом з ними. При цьому з економічної точки зору виробництво біогазу з рослинна сировини розглядається виключно як допоміжний, додаткове джерело доходу. В якості вихідного субстрату біогазового реактора в цьому випадку можуть бути використані рослинні залишки традиційних зернових культур[5].

Позитивним ефектом впровадження біогазового контуру в виробничий процес може бути отримання великої кількості високоякісного добрива, оскільки шлак, що

залишається після анаеробного зброджування в біореакторі, багатий базовими поживними сполуками. Комплексне дослідження пропонованої замкнутої технологічного ланцюжка в масштабі середнього рослинницької господарства в натурних умовах може зажадати значних тимчасових і фінансових витрат. Альтернативним підходом до дослідження проблеми може служити технологія імітаційного комп'ютерного моделювання. Створення імітаційного комплексу з інтегрованих моделей виробництва біогазу і продукційного процесу сільськогосподарських культур дозволить отримати адекватний інструментарій, що дозволяє замінити трудомісткий польовий досвід багатофакторним обчислювальних експериментів.

Важливою вимогою при до даної моделі є її багатокомпонентність. Дійсно, в умовах реального сівозміни наявна зелена біомаса, що служить сировиною для виробництва біогазу, може характеризуватися сильною неоднорідністю складу. Для того щоб включення контуру виробництва біогазу в традиційний виробниц-жавного цикл рослинницької господарства було виправданим, гіпотетичний реактор повинен бути «всеїдним» і забезпечувати ефективно і безперебійне функціонування незалежно від якості і ступеня різноманітності одночасно споживаних ресурсів. Передбачається, що одночасно або послідовно в реактор можуть бути завантажені післяжнивні рослинні залишки традиційних харчових посівів, якісна біомаса спеціально вирощуваних культур енергетичного призначення з великим вмістом лабільних цукрів. Ясно, що і енергетична цінність і швидкість розкладання різних компонентів подібної сировинної суміші дуже сильно розрізняються, що має знаходити адекватне відображення у відповідній математичній моделі процесу.

Іншим суттєвими вимогам, які ставляться до моделі, є можливість її використання для розв'язання оберненої задачі визначення значень вхідних даних і параметрів, що забезпечують необхідні або оптимальні значення вихідних характеристик. У нашому випадку це означає знаходження оптимальних режимів виробничого циклу знаходження найкращого з економічної точки зору співвідношення між виходом біометана і темпами подачі / заміни вихідного субстрату в залежності від складу сировини. Останнє передбачає необхідність

множинних розрахунків моделі для різних варіантів в автоматичному режимі, тобто проведення спеціально спроектованих комп'ютерних експериментів по оптимізації, аналізу чутливості і т.д

Анаеробна ферментація органічної сировини[6] з метою виробництва біогазу є досить складним процесом, що складається як з послідовних хімічних реакцій коли вихід попередньої реакції служить субстратом для наступної, так і з паралельних шляхів трансформації різних хімічних сполук з утворенням одного і того ж кінцевого продукту. Кожна стадія вимагає функціонування свого пулу специфічних мікроорганізмів і має характерну швидкістю протікання хімічних процесів. У будь-якому випадку для описання динаміки цього процесу повинні служити так звані стехіометричні моделі, засновані на чисельному рішенні рівнянь хімічної кінетики, включених в розгляд реакцій. Прийнято виділяти чотири головних стадії розкладання органічної сировини до біогазу. Перша стадія гідроліз високомолекулярних органічних сполук (вуглеводів, білків і жирів) - їх розкладання до складових їх полімерів (відповідно цукрів, амінокислот і жирних кислот). Тільки ці останні, більш прості сполуки, можуть піддаватися впливу специфічних мікроорганізмів для подальшого розкладання. Друга і третя стадія називаються, відповідно, ацито- і ацетогенеза. В ході ацитогенеза під дією анаеробних бактерій відбувається утворення складних спиртів і карбонових кислот, які на третій стадії ацетогенеза розкладаються до оцтової та мурашиної кислоти, а також метанолу з виділенням водню . Нарешті, в ході четвертої стадії відбувається утворення власне біогазу. Саме на цьому етапі в дію вступають специфічні бактерії, в результаті чого з продуктів ацетогенеза утворюються вуглекислий газ і водень основні компоненти суміші, званої «біогаз»[7]. До теперішнього часу розроблено багато моделей, що описують перераховані процеси з різним рівнем детальності.

Варто зауважити, що принциповою відмінністю всіх наступних етапів від стадії гідролізу є те, що вони протікають при безпосередній участі специфічних анаеробних організмів. Крім того, основна відмінність різних видів рослинної сировини з точки зору його ефективності для виробництва біогазу проявляється саме в темпах гідролізу вихідної сировини, так як наступні стадії анаеробного

бродіння в будь-якому випадку проходять над практично одними і тими ж вторинними субстратами.

Виходячи з цих міркувань, в більшості математичних моделей прикладної спрямованості вважається допустимим без істотної втрати точності обмежити розгляд описом трьох основних процесів[8]:

- гідроліз (первинне розкладання) вихідного субстрату (реакція, в першому наближенні, що протікає без залучення зовнішніх каталізаторів або ферментів);
- метаногенеза - утворення біогазу з продуктів первинного розкладання органічної сировини під дією специфічних анаеробних мікроорганізмів таким чином, в опис узагальненого процесу метаногенеза неявно включаються також стадії ацито- і ацетогенеза);
- динаміка зростання і розкладання самої мікробної біомаси, необхідної для протікання процесів анаеробного бродіння.

Більшість прикладних задач, що виникають у виробництві, пов'язаному з технологією хімічної переробки органічної сировини, вимагають для свого дослідження побудови адекватних математичних моделей досліджуваного процесу. Як правило, в даній предметній області все зводиться до моделей хімічної кінетики, або так званих стехіометричних моделей, математичною формою яких виступають звичайні не лінійні диференціальні рівняння. Виключно зручним інструментом комплексного дослідження подібних моделей системної динаміки можуть служити сучасні високопродуктивні і високорівневі середовища імітаційного моделювання. Вбудовані в них стандартні процедури параметричної ідентифікації, аналізу чутливості та стійкості, оптимізації та статистичної обробки результатів дозволяють ефективно вирішувати завдання чисельного дослідження досліджуваного процесу і знаходити найкращі режими функціонування технічних систем

2.1 Попередня обробка целюлозовмісної сировини

Ферментативна конверсія відходів з метою одержання енергоносіїв залежить від параметрів анаеробного процесу та якісного і кількісного складу сировини. Як сировину можна використовувати відходи сільськогосподарських рослин та деревообробної промисловості. Хімічний склад сировини залежить від виду рослин, умов вирощування та зберігання[9]. Так, в залежності від виду сільськогосподарської сировини солома містить: 22 ÷ 49 % вуглеводів, які важко піддаються гідролізу, 16 ÷ 26% лігніну, який дуже повільно піддається деструкції мікроорганізмами. Деревина містить лігніну до 30% і до 56% вуглеводів, що важко гідролізуються. При цьому лігнін є клійковиною, що цементує волокна целюлози між якими знаходяться геміцелюлози та інші екстрактні речовини. Міжклітинною речовиною, в основному, є пектати кальцію та магнію. Клітини рослин також містять жири, вуглеводи, сапоніни, флавоноїди, каратини тощо. Для процесу одержання енергоносіїв з такої сировини з високою швидкістю їх утворення необхідно збільшити доступ мікроорганізмів до живильних речовин.

Для збільшення місць дотику ферментів, що знаходяться на поверхні клітин, до поверхні твердої сировини запропоновано проводити попередню обробку. Попередня обробка дозволяє збільшити розміри пор і зменшити ступінь кристалічності целюлози для доступності ферментів. Одночасно відбувається знешкодження метаногенних бактерій.

Як відновлювану сировину можна використовувати відходи пшениці, ячменю, кукурудзи, ріпаку, соняшнику, деревини берези, сосни. Вибір сировини було зроблено виходячи з масштабів вирощування культур. Ячмінь – друга після озимої пшениці у валовому зборі сільськогосподарська культура на Україні. Вирощування ріпаку щороку зростає і на сьогодні в Україні займає 1058 тис. га. Основними породами дерев лісів в Україні є сосна, ялиця, бук, дуб, липа, клен та береза.

2.1.1 Попередня обробка сировини парою

Попередню обробку подрібненої сировини парою проводили протягом 1 год. без застосування тиску. У табл. 2.1 наведено вихід водню, що утворюється при ферментації різних видів відновлюваної сировини, на 10 добу попередньо обробленої паром (1 год.) та необробленої сировини. Для запобігання впливу парціального тиску водню, що утворюється, на метаболізм мікроорганізмів, щодобово відводили газ з зони реактора. Вміст компонентів газу розраховували в залежності від їх співвідношення в газовій суміші, приймаючи за 100% максимальний об'єм газу, що утворився у зразку з соломою ячменю[10].

Таблиця 2.1

Вихід водню при конверсії відходів різного походження після попередньої обробки парою та без неї

Сировина	pH поч./кінц.	H ₂ , % оброб.	CO ₂ , % оброб.	H ₂ , % необроб.	CO ₂ , % необроб.
Солома пшениці	7,2/6,8	16,4	33,1	1,4	3,1
Солома ячменю	7,0/6,4	54,3	42,6	6,1	9,4
Кукурудза (стебла, листя)	7,1/6,6	41,2	38,7	4,8	7,9
Соняшник (стебла, голівки)	7,0/6,9	11,1	21,5	1,3	2,8
Солома ріпаку	6,8/7,0	1,7	3,6	0,3	0,8
Тирса сосни	6,7/7,15	0,65	2,1	0	0,1
Тирса берези	7,0/7,0	1,13	2,5	0,1	0,2

Як видно з табл. 2.1. застосування обробки сировини парою підвищує швидкість утворення водню і, відповідно, вихід біогазу. На вихід водню впливає компонентний склад сировини. Найбільший вихід водню спостерігається для соломи ячменю, в якій міститься найменша кількість лігніну (11%) і найбільша

кількість полісахаридів, що легко гідролізуються (44%). Збільшення кількості лігніну і зменшення кількості полісахаридів, що легко гідролізуються, призводить до зниження швидкості розкладу целюлозовмісної сировини і продукування водню, як це характерно для соломи кукурудзи, пшениці та соняшника. Як видно з табл. 2.1 швидкість перетворення природної сировини більша, ніж чистої целюлози. Деструкція чистої целюлози відбувається майже у тричі повільніше, ніж соломи ячменю. Пояснити це можна тим, що солома ячменю містить значну кількість геміцелюлози – полісахаридів із значно коротшими ланцюгами, що розчинні у воді. Завдяки цьому збільшується кількість доступних живильних речовин для утворення водню. Також, солома ячменю містить удвічі менше протеїнів по відношенню до інших видів сировини, що також впливає на швидкість продукування водню. Мікроорганізми для продукування водню використовують вільні амінокислоти, при чому в асоціації кількість видів, які використовують амінокислоти як субстрат, значно менша, ніж видів, що використовують цукри. При цьому солома ячменю містить інші легкодоступні низькомолекулярні речовини, розклад яких забезпечує утворення водню.

На мікробіологічний розклад біоенергетичної сировини впливає також наявність в ній мінеральних солей[11]. Зразки деревини, окрім значної кількості лігніну та полісахаридів, що важко гідролізують, містять незначну кількість мінеральних солей, нітрогену, сульфуру та фосфору, тобто елементів, які необхідні для розвитку мікроорганізмів. Також в цьому випадку, враховуючи густину деревини – 510 кг/м³ для сосни і 640 кг/м³ для берези (40 кг/м³ для соломи пшениці), термін її обробки у такому режимі може бути недостатнім, а збільшення терміну обробки парою є енергетично не вигідним у масштабах промислового виробництва.

При катаболізмі органічної сировини біоводень, в основному, утворюється в анаеробному процесі метаболізму пірувату. Утворення інших продуктів перетворення (кислот – масляна, оцтова, молочна; спиртів – етанол, бутанол, 2-пропанол; а також ацетону) та їх співвідношення залежить від складу сировини, умов культивування та віку культури. За даними хроматографічного аналізу при

старінні культури відбувається перехід з масляного типу бродіння на ацетоно-бутилове, що супроводжується зменшенням виходу водню після 20 доби.

Таким чином, обробка біоенергетичної сировини парою підвищує швидкість розкладу сільськогосподарської біоенергетичної сировини і, відповідно, вихід водню у $6 \div 10$ разів по відношенню до необроблених відходів. На продукування водню мікроорганізмами впливає компонентний склад сировини, а саме вміст лігніну, полісахаридів, що гідролізуються, та мінеральних солей. Тривалість процесу також впливає на вихід водню. При старінні культури відбувається зміна метаболізму з переходом на утворення таких продуктів як етанол та бутанол, і відповідно, знижується вихід водню. З метою одержання водню з відходів ячменю та кукурудзи пропонується застосовувати попередню обробку парою.

2.1.2 Попередня обробка сировини лугом

Відходи різного походження кип'ятили протягом 2 год. у розчині $0,5 \div 10$ моль/дм³ NaOH. Виміри проводили за 10 діб ферментації. Для вилучення карбон (IV) оксиду та H₂S біогаз пропускали через розчин лугу.

У табл. 2.2 наведено вихід водню, що одержано в анаеробному мезофільному процесі ферментації, з різної сировини при 2 годинах її попередньої обробки лугом.

Таблиця 2.2

Вихід водню після попередньої обробки відходів NaOH протягом 2 год.

Сировина	Вихід водню, см ³				
	0,5 M NaOH	1,0 M NaOH	2,0 M NaOH	5,0 M NaOH	10,0 M NaOH
Солома ячменю	58 ± 2	65 ± 2	74 ± 1	75 ± 1	75 ± 1
Солома пшениці	27 ± 2	33 ± 2	37 ± 2	39 ± 2	39 ± 2
Кукурудза	57 ± 2	69 ± 2	74 ± 1	75 ± 1	75 ± 1
Солома ріпаку	$4 \pm 0,5$	9 ± 1	11 ± 1	19 ± 1	$20 \pm 1,5$
Соняшник	17 ± 1	$24 \pm 1,5$	30 ± 2	32 ± 2	32 ± 2
Деревина сосни	$1 \pm 0,5$	$3 \pm 0,5$	$8 \pm 0,5$	10 ± 1	10 ± 1
Деревина берези	$1,8 \pm 0,5$	$4 \pm 0,5$	$8,5 \pm 0,5$	11 ± 1	11 ± 1

Під дією лугу відбувається виділення лігніну[12], що значно полегшує надходження живильних речовин до клітин мікроорганізмів. Саме цим можна пояснити збільшення виходу водню за використання соломи пшениці та соняшнику по відношенню до значень при їх попередній обробці паром. Відділенням лігніну можна пояснити незначне підвищення виходу водню і при використанні деревини. Дані таблиць свідчать, що за використання обробки сировини лугом для досягнення максимального виходу водню необхідно збільшувати термін обробки.

Для пшениці найбільший вихід водню відбувається при попередній обробці лугом. Тобто, обробка лугом є ефективною при вмісті лігніну у сировині більше 20%. Використання такої обробки зменшується термін лаг-фази і збільшується вихід водню.

Одержані дані дозволяють встановити методи попередньої обробки відновлюваної сировини в залежності від її компонентного складу для одержання біоводню або біометану. Для сировини, що містить лігніну до 20%, більше 30% полісахаридів та інших речовин, що легко гідролізуються, та у достатній кількості мінеральних солей ($5 \div 8\%$), що забезпечують розвиток мікроорганізмів мікроелементами, а також достатню кількість нітрогену та фосфору необхідно застосувати попередню обробку парою. Сировині такого складу відповідають солома ячменю та кукурудзи. Така обробка не потребує додаткових витрат на хімічні речовини, не привносить інгібуючих компонентів, не змінює рН і дозволяє досягти максимального виходу водню. Єдиною умовою для використання соломи ячменю та кукурудзи як сировини для одержання водню є відсутність обробки рослин при вирощуванні пестицидами, що мають інгібуючий вплив на розвиток мікроорганізмів. При використанні соломи ячменю як субстрату максимальний вихід водню складав 48 г H_2 / кг сухої біомаси на 10 добу конверсії асоціацією мікроорганізмів, концентрація водню в біогазі досягає 54%.

У випадку використання сировини, що містить лігніну більше 20%, геміполісахаридів менше 30% необхідно використовувати обробку лугом. При цьому тривалість обробки та концентрація лугу необхідно збільшувати при підвищенні концентрації лігніну. Обробка лугом призводить до руйнування

лігніноцелюлозного каркасу сировини і дозволяє вивільнити та осадити лігнін, що полегшує доступ мікроорганізмів до волокон целюлози та інших живильних речовин. До такої сировини відноситься солома пшениці, соняшника, ріпаку. Концентрація водню в газі, що утворився при переробці соломи пшениці, досягає 35%.

Використання деревини в чистому виді для одержання водню мікробіологічним шляхом не є рентабельним, оскільки деревина не містить достатньої кількості мінеральних компонентів, які необхідні для розвитку мікроорганізмів. Тому для її використання у ферментативних процесах необхідно додавати мінеральні солі або інші компоненти з їх вмістом, наприклад, гній. Але в останньому випадку для запобігання утворенню метану процес необхідно проводити в кислому середовищі, що зменшує швидкість процесу утворення водню і, відповідно, його вихід.

2.2 Отримання водню з целюлозовмісної сировини у анаеробному процесі

Ферментативний анаеробний процес отримання енергоносіїв поділяється на дві стадії:

1. деструкція високомолекулярних речовин і утворення органічних кислот;
2. утворення з низькомолекулярних речовин енергоносіїв.

За використання сільськогосподарських відходів найбільша швидкість продукування водню відбувається за нейтральних значень рН. Найбільша швидкість процесу деструкції – при $\text{pH} = 4 \div 6$. Лімітуючою стадією є повільний розклад сировини, який залежить від гідролітичної активності мікроорганізмів. Оскільки процеси деструкції сировини та утворення енергоносіїв відбуваються за різних оптимальних умов, тому розділення процесу ферментації на стадії дозволяє створити необхідні умови для утворення консорціуму мікроорганізмів, характерних для процесу, якій перебігає на кожній стадії.

Одержання біоводню можна проводити в одну стадію[13]. В такому випадку процес проводять або при рН $5 \div 6$ для збільшення швидкості деструкції сировини, або за нейтральних значень рН для збільшення швидкості продукування водню. Низькі значення рН розчину дозволяють також знешкодити метаногенні мікроорганізми, але знижують вихід водню. У випадку одержання біогазу процес необхідно проводити за значень рН, що відповідають утворенню біогазу ($6,5 \div 7,5$).

2.3 Вплив параметрів процесу конверсії відходів на вихід водню

Утворення водню в метаболізмі мікроорганізмів залежить від багатьох факторів: парціального тиску водню, якісного складу живильних речовин, кислотності середовища, наявності мікроелементів тощо.

В процесі утилізації органічної сировини угрупованням мікроорганізмів утворюються, окрім CO_2 та H_2 , органічні кислоти і спирти, при цьому також утворюються багаті на енергію речовини, які забезпечують біосинтез необхідних клітині речовин.

Ферментна система мікроорганізмів інгібується кінцевими продуктами біосинтезу. Окрім органічних кислот, що утворюються в процесі продукування водню, розчинений CO_2 також призводить до підвищення кислотності середовища, що буде зміщувати метаболічні шляхи в напрямку синтезу таких субстратів як лактат, етанол, ацетон, бутанол або аланін, які знижують кислотність. Зміна метаболічних шляхів з продукування кислот на нейтральні продукти знижує вихід водню.

Кількість CO_2 та інших кислот, що утворюються в процесі ферментації, залежить як від концентрації сировини, так і кількості мікроорганізмів у реакторі. Як інокулят використовували асоціацію мікроорганізмів з компосту. Вміст інокуляту – 0,2 дм³. Сировина – відходи кукурудзи попередньо подрібнені і оброблені парою 1 год. Об'єм заповненого реактора – 1,2 дм³. Процес проводили в мезофільному режимі при температурі $35 \pm 50^\circ\text{C}$ при перемішуванні $10 \div 100$ об./хв.

Для зниження життєдіяльності метаногенних мікроорганізмів запуск реактора проводили при наявності повітря. Біогаз, що одержували, пропускали через розчин лугу.

Таким чином, при проведенні ферментативного процесу одержання водню за використання твердої відновлюваної сировини в одну стадію необхідно враховувати співвідношення кількості мікроорганізмів і субстрату. За вмісту інокуляту 1/6 об'єму (17 г сухої біомаси/дм³) заповненої зони реактора вміст відходів за сухою речовиною не повинен перевищувати 50 г/дм³. Оскільки при збільшенні кількості сировини зменшується значення рН, що призводить до зниження виходу водню і зміни видів мікроорганізмів та їх співвідношення в асоціації. При цьому утворені низькомолекулярні жирні кислоти проникаючи до клітин змінюють рН внутрішньоклітинного середовища, що призводить до інгібування гідрогенази та відсутності активації генів, що відповідають за її біосинтез.

Висновки до розділу

У розділі було розглянуто імітаційну модель процесу виробництва водню з багатокомпонентної рослинної сировини. Досліджено основні вимоги до імітаційної моделі процесу. Було розглянуто три основні процеси утворення водню з продуктів первинного розкладання органічної сировини під дією специфічних анаеробних мікроорганізмів. Визначення залежності процесу продукування водню від компонентного складу відновлюваної сировини та методів її попередньої обробки. Було досліджено основні особливості процесу утворення водню з відновлюваної целюлозовмісної сировини у біореакторі.

3. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Для реалізації поставленої задачі треба обрати найбільш раціональний шлях для вирішення поставлених задач. У ході роботи над було проведено аналіз, який допоміг обрати технології та методи програмної реалізації для кращого виконання роботи.

Аналізуючи поставлену задачу було прийнято рішення створити програмний комплекс на основі веб-технологій. Основною перевагою веб-застосунку перед іншими варіантами є його універсальність і можливість використання на будь-яких пристроях без встановлення на цільову операційну систему. Для веб-додатку браузер і його віртуальна машина слугує цільовою універсальною операційною системою та комп'ютером.

3.1 Вибір архітектури програмного комплексу

Архітектура програмного забезпечення - це структура програмного застосунку, що складається з певних програмних компонентів, видимих зовні властивостей цих компонентів та зв'язками між ними.

Архітектуру програмного забезпечення можна уявити у вигляді розробки стратегії — визначенням глобальних обмежень та вимог, що необхідно застосувати до системи на етапі її проектування. Детальне проектування — це не що інше як визначення певних локальних обмежень для проекту. Мова йде про архітектурні моделі, шаблони проектування та ідіоми програмування.

Для реалізації поставленої задачі було вирішено використовувати триланкову архітектуру, яка складається з таких компонентів: сервер, база даних і клієнт. Схема даної архітектури зображена на рисунку 3.1.

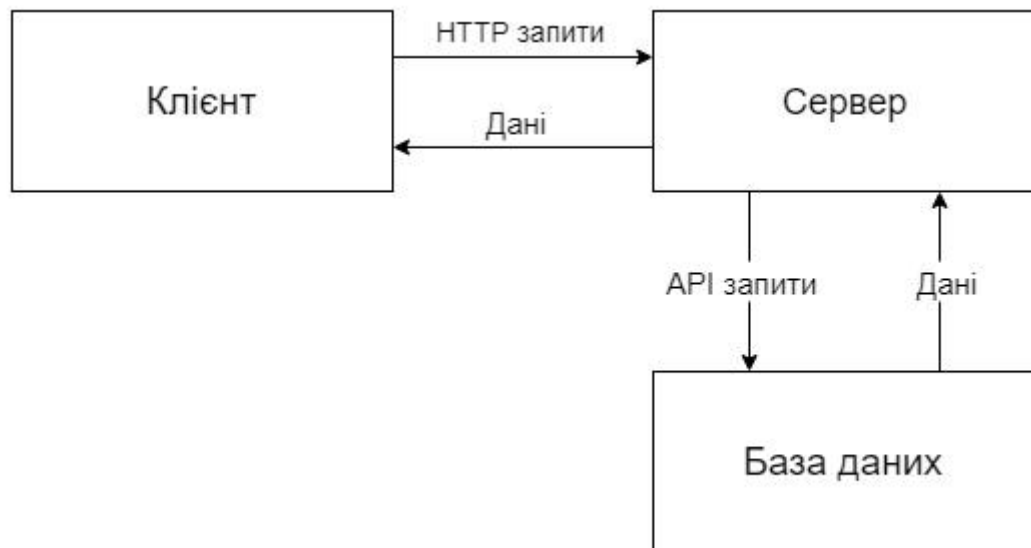


Рисунок 3.1 – Триланкова архітектура програмного комплексу

Основною функцією серверної частини є ідентифікація користувача для надання доступу до головного функціоналу веб-застосунку. Пошкодження даних про користувача, що знаходяться в базі даних є малоймовірним, тому що єдиним зв'язком між базою даних та користувачем є сервер. Такий зв'язок унеможливорює використання даних не за призначенням. Варто наголосити, що основний функціонал системи є доступний користувачу тільки після авторизації у системі. Логіка ідентифікації користувача реалізується на рівні серверу, тому що на рівні користувача може статися підміна прав доступу до системи.

Під час використання системи, користувач взаємодіє з клієнтським додатком, яким є веб-сайт в даному випадку. На рівні користувача реалізований інтерфейс, за допомогою якого відбувається введення даних про сировину та перегляд результатів роботи. Також на користувацькому рівні відбувається обробка та аналіз даних. Ще на цьому рівні відбувається перший етап аутентифікації користувача для обмеження неконтрольованого доступу до програмного застосунку.

3.2 Опис архітектури клієнтського застосунку

Для реалізації клієнтського застосунку було використано фреймворк Vue.js, в основу якого було покладено шаблон проектування MVVM (рисунок 3.2).

MVVM (Model-View-ViewModel) — це патерн, що використовується для проектування архітектури веб-додатків та інших програмних застосунків.

Шаблон MVVM складається з трьох частин:

- Модель (Model). Модель являє собою фундаментальні дані, що необхідні для роботи застосунку.
- Представлення (View). Це графічний інтерфейс користувача.
- Модель вигляду (ViewModel, що означає “Model of View”) з одного боку є абстракцією Представлення, а з іншого надає обгортку даних з Моделі, які мають зв'язуватись.



Рисунок 3.2 – Схема роботи MVVM шаблону

Шаблон проектування MVVM дозволяє відносно просто відокремити розробку графічного інтерфейсу користувача від розробки моделі, що представлена у вигляді бек-енд логіки. Інакше кажучи, MVVM реалізовує відокремлення моделі від представлення.

Під моделлю, зазвичай розуміється частина системи що містить в собі функціональну бізнес-логіку програми. Модель повинна бути повністю незалежна від інших частин системи. Модельний рівень нічого не повинен знати про елементи дизайну, і яким чином вони будуть відображатися. Таким чином, стає можливою зміна представлення даних, те як вони відображаються, що не залежить від зміни моделі.

Модель володіє наступними ознаками:

- модель - це бізнес-логіка програми;
- модель володіє знаннями про себе саму і не знає про контролери та представлення;
- для деяких проектів модель - це просто рівень даних (DAO, база даних, XML-файл);
- для інших проектів модель - це менеджер бази даних, набір об'єктів або просто логіка застосунка

В обов'язки представлення входить відображення даних отриманих від моделі. Однак, уявлення не може безпосередньо впливати на модель. Можна говорити, що уявлення отримує доступ «тільки на читання» до даних.

Представлення володіє наступними ознаками:

- у представленні реалізується відображення даних, які виходять від моделі будь-яким способом;
- у деяких випадках, представлення може мати код, який реалізує деяку бізнес-логіку.

3.3 Опис інструментів розробки

Програмний комплекс побудований за принципами триланкової архітектури побудови програм. Кожен рівень цієї архітектури реалізовано з використанням різних технологій і головною ціллю було створення мультиплатформного рішення з використанням відкритих технологій.

Для клієнтського рівня було використано такий набір технологій: мова програмування JavaScript, фреймворк для побудови веб-застосунків Vue.js, бібліотеку для візуалізації даних chart.js, стандартна мова розмітки для створення веб-сторінок і веб-додатків HTML5, скриптова метамова Sass

3.3.1 Мова програмування JavaScript

JavaScript - це легка об'єктно-орієнтована мова програмування з функціями першого класу, найвідоміша скриптова мова для веб-сторінок, але також використовується у багатьох не браузерних середовищах. JavaScript являє собою прототипно-орієнтовану, мультипарадигмальну мову сценаріїв, яка підтримує динамічний, об'єктно-орієнтована, імперативний і функціональний стилі програмування.

JavaScript запускається на стороні клієнта Інтернету, який може використовуватися для створення сценарії поведінки веб-сторінки при настанні будь-яких подій. JavaScript - потужна скриптова мова, що широко використовується для контролювання поведінки веб-сторінок.

JavaScript використовується практично на всіх сучасних веб-сайтах. У всіх сучасних браузерах і клієнтських пристроях - настільних комп'ютерах, ігрових консолях, планшетах і смартфонах - є інтерпретатори JavaScript, що робить його найбільш поширеною мовою в історії програмування. Він входить в базову тріаду технологій, які необхідно знати всім розробникам веб-додатків.

3.3.2 Фреймворк Vue.js

Vue.js - це JavaScript бібліотека для створення веб-інтерфейсів з використанням шаблону архітектури MVVM[14] (Model-View-ViewModel).

Vue.JS створювався з урахуванням найкращих практик вже існуючих технологій. З React.JS фреймворк Vue запозичив ідею віртуального DOM. Цей підхід виключає пряму взаємодію з вузлами інтерфейсу. Початкова робота ведеться з його легкою копією (virtual DOM). І тільки після цього зміни застосовуються до реальних вузлів інтерфейсу. Паралельно відбувається порівняння реального DOM дерева і його віртуальної копії. Таким чином виявляється різниця і змінюється тільки те, що зазнало змін.

З Angular Vue.JS запозичив two-way data binding. Це дозволяє проектувати інтерфейси декларативно та з використанням Vue в таких шаблонізаторах як Haml або Pug.

Ядро Vue.JS, подібно до React, оскільки містить лише необхідний функціонал для роботи з інтерфейсом[15]. Тому воно компактне, легко інтегрується з іншими технологіями, в тому числі з jQuery і навіть може використовуватися замість нього (для розробки простих інтерфейсів).

Крім того, у Vue доступний ряд модулів, що реалізують сучасний підхід до розробки веб-додатків. Для Vue.JS була розроблена своя технологія контролю стану програми - Vuex. Vuex повністю запозичує ідеї Redux, але ступінь інтеграції цієї бібліотеки з Vue набагато вище, ніж в разі React і Redux.

Vue підходить для невеликих проектів, яким необхідно додати трохи реактивності, реалізувати форму за допомогою AJAX, відобразити значення при введенні даних користувачем, авторизація або інші аналогічні завдання. Vue легко масштабується і добре підходить для об'ємних проектів, тому його називають прогресивним фреймворком.

Vue також відмінно підходить для великих односторінкових додатків завдяки своїм основним компонентам, таким як Router і Vuex. З Vue можна як використовувати загальнодоступні API для створення додатків, так і реалізовувати виконуються сервером додатка. Але Vue найкраще підходить для розробки рішень, які використовують зовнішні API для обробки даних.

3.3.3 Серверна платформа Node.js

Node.js — платформа з відкритим кодом, що використовується для створення клієнтських та серверних застосунків[16].

Один з найважливіших плюсів платформи - це асинхронність в поєднанні з подієвим підходом. На відміну від потокового, подієво-орієнтоване програмування засноване на зовнішніх діях. Тобто виконання програми залежить від дій

користувача. Таке рішення спрощує програмування інтерактивних додатків, тобто заснованих на роботі з використанням введення-виведення. Це можуть бути онлайн-чати або ігрові програми, барвисті інтерактивні веб-сервіси або всілякі калькулятори, рейтинги, голосування і т.д.

Крім того, до переваг платформи відносять:

- Простий і широко відомий JavaScript. Звичайно, платформа передбачає власні інструменти та особливості, наприклад, тут немає браузерних API, cookie або DOM, однак присутні власні бібліотеки. В основному використовуються можливості і синтаксис всім звичного JavaScript.
- Велика стандартна бібліотека. Платформа від самого початку мала широкий набір можливостей, а в нових версіях бібліотека поповнюється і поліпшується.
- Велика кількість зовнішніх бібліотек і готових модулів. Використання пакетного менеджера NPM дозволяє постійно розвивати екосистему Node. Сьогодні число пакетів в ньому перевищило за цифру 500 тисяч і постійно зростає.

Важливо розуміти, що на відміну від звичайного браузерного JavaScript, із застосуванням Node додаток може звертатися до широкого переліку глобальних об'єктів, в тому числі document або window. В результаті програма може звертатися до вінчестера і файлової системи користувача, а також до бібліотек і програмним рішенням, що знаходяться на комп'ютері, що помітно розширює можливості взаємодії.

Найчастіше цю платформу використовують для створення веб-сервісів, що вимагають інтенсивного обміну інформацією з користувачами, в тому числі, для реалізації чатів, систем спільної роботи, соціальних мереж і т.д. Багато програм, створені на Node.js, складаються з серверної і клієнтської частин.

Для розуміння причин такого використання варто трохи докладніше зупинитися на асинхронності і можливостях подієво-орієнтованої парадигми. Суть програмування при цьому зводиться до того, що ви вказуєте програмі, які дії вона

повинна виконати, а також яку функцію вона повинна викликати після виконання цих дій. Поки Node.js чекає виконання коду і виклику функції, вона може виконувати інші завдання. При цьому потік в очікуванні запиту блокуватися не буде.

Таким чином, програма може звернутися із запитом до бази даних сервера, а поки очікується відповідь, обробити інші запити. В результаті одночасно обробляються тисячі з'єднань, кожне з яких витрачає мало часу і ресурсів. Це помітно прискорює обмін інформацією з сервером і, як підсумок, роботу браузерного додатку.

3.3.4 Фреймворк EXPRESS

Express.js, або просто Express, каркас web-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT. Він спроектований для створення веб-додатків і API. Де-факто є стандартним каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описує його як створений на основі написаного на мові Ruby каркаса Sinatra, маючи на увазі, що він мінімалістичний і включає велику кількість додаткових плагінів. Express може бути backend'ом для програмного стека MEAN, разом з базою даних MongoDB і каркасом Vue для фронт-енду[17].

3.3.5 База даних MongoDB

MongoDB є відкритою системою управління базами даних (СКБД), яка використовує документно-орієнтовану модель бази даних, що підтримує різні форми даних[18]. Це одна з численних не реляційних технологій баз даних, які виникли в середині 2000-х років під банером NoSQL для використання у великих додатках та інших місцях обробки, що містять дані, які не вписуються в жорстку реляційну модель. Замість використання таблиць і рядків, як у реляційних базах даних, архітектура MongoDB складається з колекцій і документів.

Запис в MongoDB є документом, який є структурою даних, що складається з пар поля-значення. Документи MongoDB подібні до об'єктів JavaScript Object Notation, але використовують варіант Binary JSON (BSON), який вміщує більше типів даних. Поля в документах близькі до стовпців реляційної бази даних, а значення, які вони містять, можуть бути різними типами даних, включаючи інші документи, масиви документів, згідно з керівництвом користувача MongoDB.

Документи, які також повинні включати первинний ключ як унікальний ідентифікатор, є основною одиницею даних у MongoDB. Колекції містять набори документів і функціонують як еквівалент реляційних таблиць бази даних. Колекції можуть містити будь-який тип даних, але обмеження полягає в тому, що дані в колекції не можуть бути розподілені між різними базами даних.

Оболонка mongo - це інтерактивний інтерфейс JavaScript для MongoDB, який дозволяє користувачам запитувати і оновлювати дані, а також проводити адміністративні операції. Оболонка є стандартним компонентом розподілів з відкритим вихідним кодом MongoDB. Після встановлення MongoDB користувачі підключають оболонку mongo до запущених MongoDB-екземплярів.

Формат зберігання та обміну даними BSON, що використовується в MongoDB, забезпечує двійкове представлення документів, подібних до JSON. Автоматична функція sharding - це ще одна ключова особливість, яка дозволяє розподілити дані в колекції MongoDB по декількох системах для горизонтальної масштабованості, оскільки обсяги даних і вимоги до пропускну здатності зростають.

СКБД NoSQL використовує єдину архітектуру майстра для узгодженості даних, з вторинними базами даних, які підтримують копії первинної бази даних. Операції автоматично копіюються до цих вторинних баз даних для автоматичного переключення. Найновіші технології MongoDB орієнтовані на хмарні та мобільні пристрої.

Як і інші бази даних NoSQL, MongoDB не вимагає попередньо визначених схем і зберігає будь-які типи даних. Це дає користувачам можливість створювати будь-яку кількість полів у документі, що полегшує масштабування баз даних MongoDB у порівнянні з реляційними базами даних.

Однією з переваг використання документів є те, що ці об'єкти відображаються у нативні типи даних у ряді мов програмування. Крім того, наявність вбудованих документів зменшує потребу в об'єднанні баз даних, що може зменшити витрати.

Основною функцією MongoDB є його горизонтальна масштабованість, що робить її корисною базою даних для компаній, що працюють з великими додатками даних. Крім того, sharding дозволяє базі даних поширювати дані по кластеру машин. Нові версії MongoDB також підтримують створення зон даних на основі клавіші shard. MongoDB підтримує ряд механізмів зберігання даних і надає інтерфейси API, які дозволяють третім сторонам розробляти власні механізми зберігання для MongoDB.

СКБД також має вбудовані можливості агрегації, які дозволяють користувачам запускати код MapReduce безпосередньо в базі даних, а не запускати MapReduce на Hadoop. MongoDB також має власну файлову систему GridFS, подібну до Hadoop Distributed File System (HDFS), в першу чергу для зберігання файлів, розмір яких перевищує обмеження розміру BSON на 16 МБ на документ. Ці подібності дозволяють використовувати MongoDB замість Hadoop, хоча програмне забезпечення бази даних інтегрується з Hadoop, Spark та іншими фреймворками обробки даних.

Хоча переваг багато, є деякі недоліки для MongoDB. Завдяки автоматичній стратегії відмови користувач встановлює лише один головний вузол у кластері MongoDB. Якщо майстер виходить з ладу, підлеглий вузол автоматично перетворюється на новий. Цей перемикач забезпечує безперервність, але не миттєву - цей процес може тривати до однієї хвилини. Для порівняння, база даних Cassandra NoSQL підтримує кілька головних вузлів. Один головний вузол MongoDB також обмежує швидкість запису даних у базу даних. Дані запису повинні бути записані на майстер і запис нової інформації в базу даних обмежений ємністю цього головного вузла. Інша потенційна проблема полягає в тому, що MongoDB не забезпечує повну цілісність посилання за допомогою обмежень зовнішнього ключа, які можуть вплинути на узгодженість даних. Крім того, аутентифікація користувача за

замовчуванням не включена в базах даних MongoDB, що відповідає популярності розробників за технологією.

MongoDB доступний у спільноті та комерційних версіях через постачальника MongoDB Inc. MongoDB Community Edition є відкритим випуском, в той час як MongoDB Enterprise Server надає додаткові функції безпеки, механізм зберігання в пам'яті, функції адміністрування та аутентифікації та можливості моніторингу через Ops Manager.

Графічний інтерфейс користувача (GUI) під назвою MongoDB Compass надає користувачам можливість працювати з структурою документів, виконувати запити, індексні дані та багато іншого. MongoDB Connector для BI дозволяє користувачам підключати базу даних NoSQL до своїх засобів бізнес-аналітики для візуалізації даних і створення звітів за допомогою SQL-запитів.

Слідом за сторонами інших постачальників баз даних NoSQL, MongoDB Inc. запустила в 2016 році базу даних у хмарі як сервіс під назвою MongoDB Atlas. Atlas працює на AWS, Microsoft Azure та Google Cloud Platform. Зовсім недавно MongoDB випустила платформу Stitch для розробки додатків на MongoDB Atlas, з планами розширити її на локальні бази даних.

3.3.6 Мова розмітки HTML та метамова Sass

HTML (HyperText Markup Language) - це мова гіпертекстової розмітки тексту. Гіпертекстовим HTML називається тому, що з його допомогою на сторінці можна встановлювати посилання на всі інші веб-документи. Під гіпертекстом в цьому випадку розуміється текст, пов'язаний з іншими текстами зазначених джерел.

HTML являє собою досить простий набір кодів, які описують структуру документа. HTML дозволяє виділити в тексті окремі логічні частини (заголовки, абзаци, списки і т.д.), помістити на веб-сторінку впорядковану фотографію або картинку, організувати на сторінці посилання на інші документи.

HTML не задає конкретні і точні атрибути форматування документа. Конкретний вид документа остаточно визначає лише браузер на комп'ютері користувача мережі Інтернет.

Варто відзначити, що HTML не є мовою програмування, але веб-сторінки можуть включати в себе вбудовані програми-скрипти на мові Javascript і програми-аплети написані на мові Java.

Основними компонентами HTML є:

- Тег. Тег HTML це компонент, який дає команду браузеру виконати певну задачу типу створення абзаца або вставки зображення.
- Атрибут (або аргумент). Атрибут HTML змінює тег. Наприклад, можна виправити абзац або зображення всередині.
- Значение. Значення присвоюють атрибутам відповідні зміни. Наприклад, якщо для тегу використовується атрибут вирівнювання, то можна вказати значення цього атрибута. Значення можуть бути текстовими та числовими.

Теги представляють собою зарезервовані послідовності символів, що починаються з (знаком менше) і закінчуються (знаком більше)[19].

Закриття теги відрізняється від відкритій тільки наявності символу '/'.

Sass - це один з найбільш розвинених і стабільних CSS препроцесорів, а також один з найпопулярніших препроцесорів у професіоналів.

Основними перевагами Sass є:

- сумісність з різними версіями CSS, завдяки якій ви можете використовувати будь-які CSS бібліотеки;
- велика кількість різноманітних функцій;
- Sass - це один з найстаріших CSS препроцесорів, що увібрав великий досвід за довгі роки свого існування;
- синтаксис. Існує два синтаксиси, спрощений (SASS) і розгорнутий CSS-подібний (SCSS).

3.3.7 Формат передачі даних JSON

JSON - це текстовий формат обміну даними, заснований на мультипарадигмальній мові програмування. Його основне призначення полягає у зберіганні і передачі структурованого потоку інформації. За допомогою простих правил формування конструкцій із символів в JavaScript, людина може забезпечити легкий і надійний спосіб зберігання будь-якого виду інформації. Це може бути звичайне число, цілі рядки або величезна кількість різних об'єктів, виражених в простому тексті. Крім цього, формат JSON[20] використовується для об'єднання між собою об'єктів і структури даних у вигляді набору компонентів, формуючи тим самим програмні одиниці, що дозволяють зберігати і обробляти складні записи, що складаються з декількох змінних різного типу.

Після того як файл створений, рядки що містяться в ньому досить легко перенаправляти в просторі Інтернет через будь-які шляхи передачі даних. Це пов'язано з тим, що рядок являє собою звичайний текст.

Незважаючи на можливість використання практично у всіх скриптових мовах, його назва відноситься до JavaScript.

Інструмент має такі переваги:

- охоплює порівняно невеликий обсяг, компактний;
- текстовий зміст може легко створюватися і використовуватися обчислювальною технікою і людиною;
- можна без особливих зусиль перетворити в структуру практично для всіх видів формальних мов, що використовуються для створення комп'ютерних програм;
- більшість мов програмування, таких як JavaScript, Ruby, Python або PHP, наділені функціями і спеціальними інструментами для читання і редагування файлу.

У переважній більшості випадків формат JSON використовується для передачі інформації від сервера до браузера. Цей процес, як правило, відбувається в «фоновому» режимі обміну браузера з web-сервером, а доставка здійснюється за допомогою AJAX. Це обумовлюється тим, що в процесі доставки даних відсутня необхідність перезавантажувати сторінку.

У JSON типи даних підрозділяються на кілька категорій: прості і складні. До першого виду можна віднести, перш за все, текстові рядки і числа, до другого - об'єкти.

В цілому виділяють шість основних типів:

- Числовий. При цьому числа можуть бути як без знаковими цілими, так і цілими зі знаком. Зокрема, тут може міститися дробна частина і представлення дійсних чисел у вигляді дробової частини логарифма і порядку. Цей спосіб застосовується в JavaScript для всіх числових значень без винятків, проте в інших математичних бібліотеках, в яких він використовується, кодування може відбуватися з використанням зовсім інших алгоритмів.
- Довільна послідовність (рядок) символів латинського алфавіту, цифр та елементів пунктуації (від нуля і символів юнікод). Кожний наступний рядок відділяється від попереднього рядка за допомогою парного розділового знака - лапок («текст») або з використанням символу, з написанням, зворотним по відношенню до звичайного символу, косою риси.
- Літерали або константи, що включаються безпосередньо в текст. Це може бути будь-яке значення з true і false або їх аналогів.
- Масив. Він являє собою упорядкований перелік символів від нуля і далі. Кожен символ може бути представлений в будь-якій формі.
- Об'єкт. Це хаотично складений набір пар ключ-значення. Виходячи з того, що основна функція об'єктів полягає в поданні абстрактного типу даних, рекомендується, щоб ключі були унікальними.
- Пусте значення, тобто «Null».

Інтервали між символами допускаються, якщо вони будуть використані між синтаксичними одиницями. Для цього застосовуються кілька символів: звичайне відступ, горизонтальні текстові вкладки і коса риска.

3.4 Обґрунтування вибору програмної реалізації

В процесі проектуванні системи було ретельного вивчено та проаналізовано предметну область. Після детального аналізу було вирішено розроблювати програмний продукт, який заснований на веб-технологіях для використання за допомогою веб-браузера.

Технології для серверної частини застосунку були обрані на основі таких характеристик, як зручність у використанні, відкритість вихідних кодів, актуальність в наш час та можливість виконання на будь-якій операційній системі. Фреймворк Node.js надає змогу створювати програми на будь-якій операційній системі. Node.js надає величезні можливості для створення веб-серверів будь-якої складності, забезпечуючи при цьому велику швидкодію та надійність.

Фреймворк Node.js було вибрано для розробки веб застосунку через декілька причин:

- Node.js - це середовище з відкритим вихідним кодом;
- Node.js - безкоштовний фреймворк;
- Node.js працює на різних платформах (Windows, Linux, Mac OS X тощо);
- Node.js використовує JavaScript на сервері.

Завдяки мінімалізму Express.js отримуємо в своє розпорядження легкий і швидкий інструмент, який легко можна розширювати і розвивати. При цьому важливо, що вибір модулів для Express не пов'язаний ні з якими обмеженнями: ні з кількісними, ні з функціональними. В результаті, цей фреймворк забезпечує можливість вирішувати будь-які завдання, не обмежуючи його при цьому у виборі засобів.

На клієнтському рівні було обрано технології, які задовольняють такі ж умови як і серверні, але з поправкою на виконання в браузері. було Фреймворк Vue.js було оскільки він забезпечує можливість створення графічних інтерфейсів різної складності, які легко змінювати, тестувати та розширювати в подальших циклах розробки програмного забезпечення. Мова JavaScript була обрана через те, що вона гнучка, підтримується більшістю браузерів, має високу швидкість роботи та зрозумілий синтаксис.

Одне з ключових переваг даного програмного продукту - це підтримка практично всіма відомими і найпопулярнішими браузерами.

Іншими перевагами JavaScript є:

- пряме підключення скриптів до HTML коду;
- можливість запуску програм в браузері і на сервері;
- широкий вибір додаткових бібліотек та фреймворків.

Базою даних було обрано MongoDB через те, що це проект з відкритим вихідним кодом, призначений для використання в проектах з невеликою кількістю даних для зберігання. MongoDB - це система зберігання даних, заснована на принципі зберігання документів в BSON (Binary JSON) форматі. Ця база даних не потребує опису схеми таблиць.

Основні переваги бази даних:

- має розподілений доступ до даних, розташованих на декількох серверах;
- можливо паралельне вилучення даних MapReduce;
- більш швидке вилучення простих структур даних;
- може зберігати не структуровану інформацію.

Дані технології в сукупності дають змогу створити якісний та надійний продукт, що буде захищений від патентних позовів з боку розробників, оскільки всі використані технології покриті ліцензіями, що виключають таку можливість і надають доступ до вихідних кодів даних проектів.

Висновки до розділу

У даному розділі були розглянуті основні засоби розробки програмного забезпечення. Також було розглянуто основні переваги обраних інструментів розробки програмного забезпечення.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Система управління процесом утворення водню з відновлюваної целюлозовмісної сировини у біореакторі буде складатися з двох основних компонентів. Компонента Main буде являти собою кабінет користувача в якому будуть відбуватися всі розрахунки. Основною перевагою Vue.js є реактивність. Це означає, що результат роботи функції в компоненті Main буде відображатися на HTML-сторінці без її перезавантаження.

4.1 Опис функціональності системи

Програмний застосунок для визначення основних параметрів процесу утворення водню містить у собі одного головного актора – користувач системи;

На рисунку 4.1 представлена діаграма прецедентів, яка описує функції та дії актора у системі.

Функції, що доступні користувачу:

- вхід в систему;
- перегляд довідки;
- визначення параметрів процесу;
- завантаження pdf файлу з результатом роботи;
- вихід із системи.

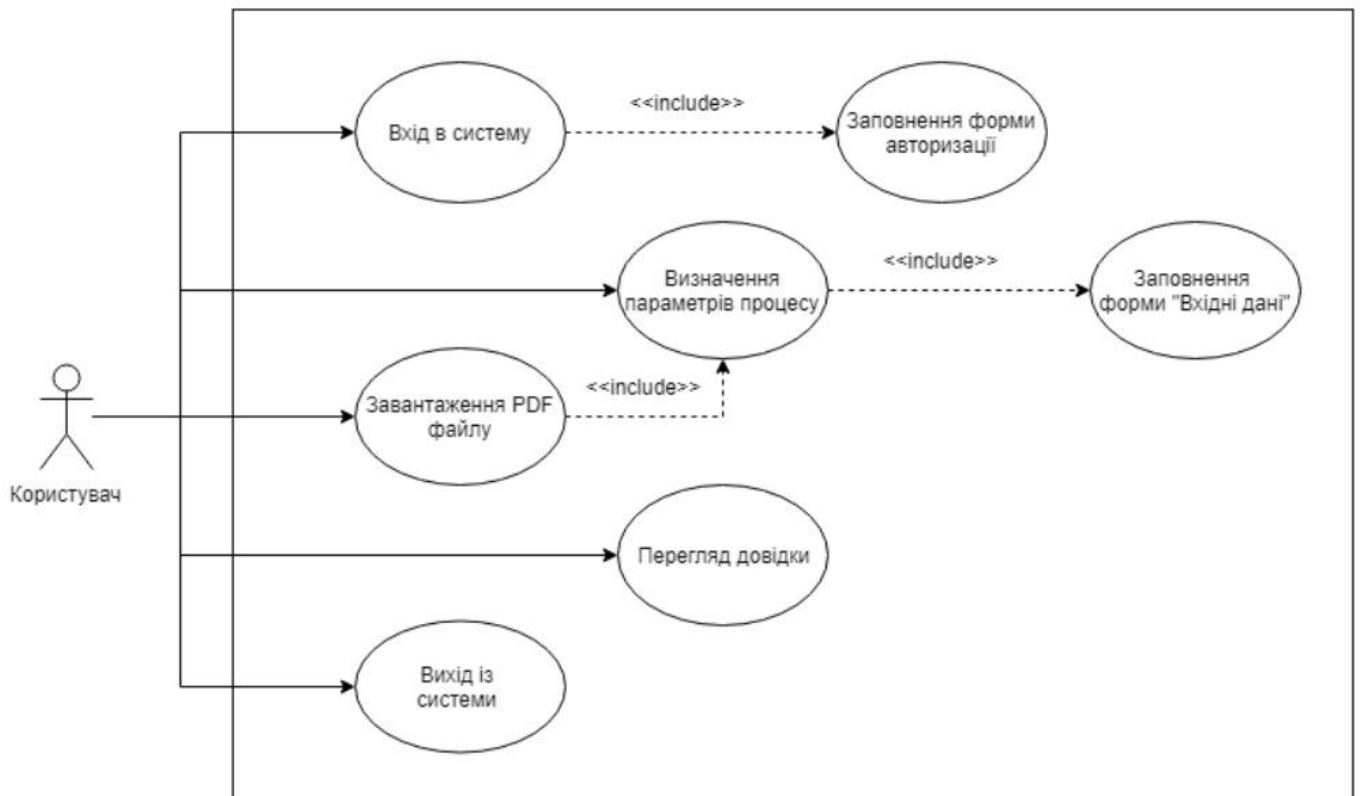


Рисунок 4.1 — Діаграма прецедентів системи

4.2 Опис бази даних

MongoDB реалізує новий підхід до побудови баз даних, де немає таблиць, схем, запитів SQL, зовнішніх ключів і багатьох інших речей, які притаманні об'єктно-реляційним базам даних.

На відміну від реляційних баз даних MongoDB пропонує документо-орієнтовану модель даних, завдяки чому MongoDB працює швидше, має кращу масштабованість, її легше використовувати.

Вся система MongoDB може представляти не тільки одну базу даних, що знаходиться на одному фізичному сервері. Функціональність MongoDB дозволяє розташувати кілька баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними і зберігати цілісність.

Одним з популярних стандартів обміну даними та їх зберігання є JSON (JavaScript Object Notation). JSON ефективно описує складні за структурою дані. Спосіб зберігання даних в MongoDB в цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання в MongoDB застосовується формат, який називається BSON (Бісон) або скорочення від binary JSON.

Якщо реляційні бази даних зберігають рядки, то MongoDB зберігає документи. На відміну від рядків документи можуть зберігати складну за структурою інформацію. Документ можна уявити як сховище ключів і значень.

Ключ являє просту мітку, з яким асоційоване певний шматок даних. У MongoDB для кожного документа є унікальний ідентифікатор, який називається `_id`. І якщо явно не вказати його значення, то MongoDB автоматично згенерує для нього значення.

Якщо в традиційному світі SQL є таблиці, то в світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити найрізноманітніші об'єкти, що мають різну структуру і різний набір властивостей.

База даних системи складається з одної колекції “users” яка містить у собі інформацію про користувача системи.

Колекції “users” приведена на рисунку 4.2.

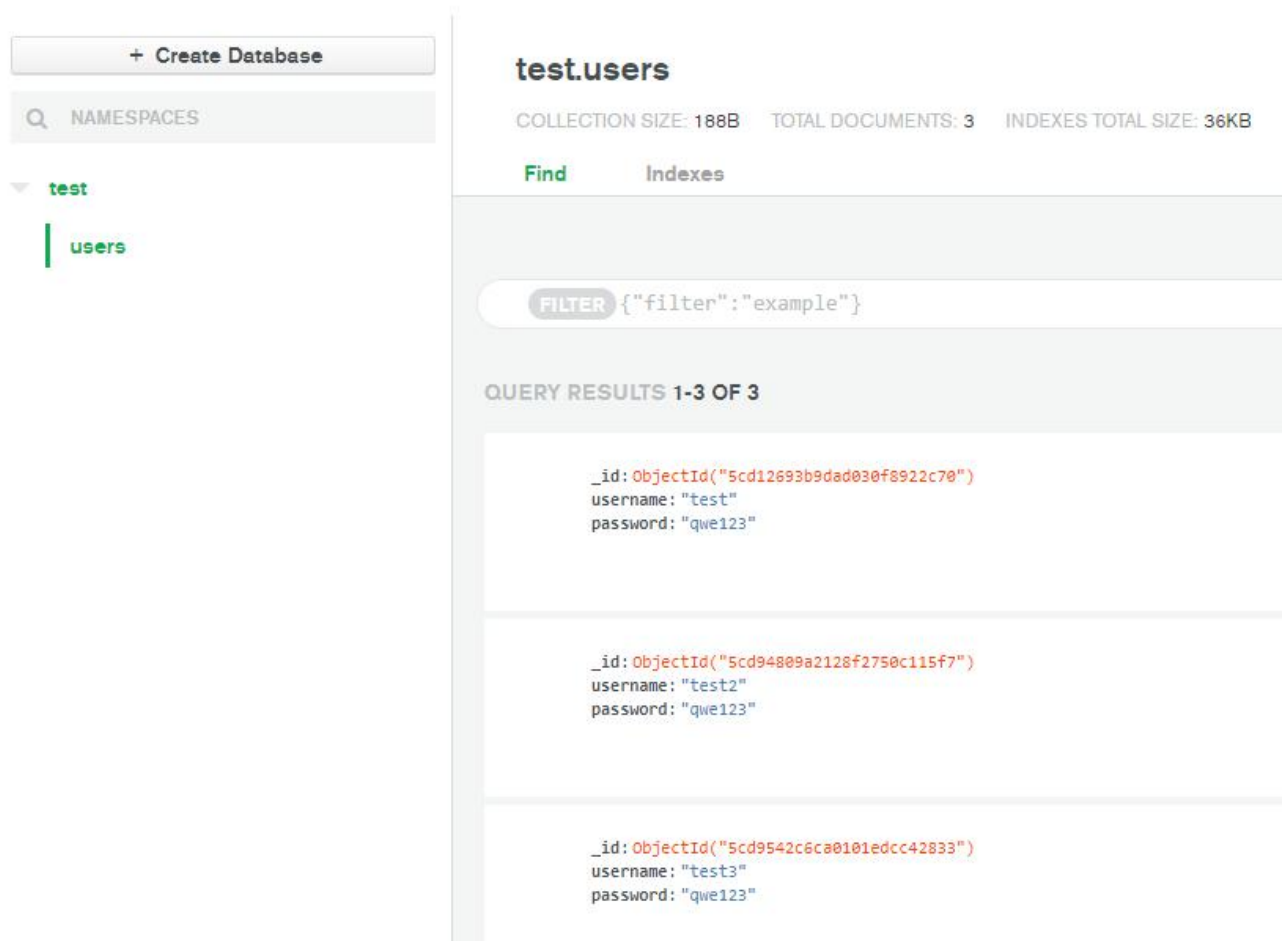


Рисунок 4.2 — Колекції “users”

Простір імен в `mongodb` це конкатенація імені бази даних та назви колекції. Як видно на рисунку, ім’я бази даних - “test”, назва колекції - “users”.

Підключення до бази даних реалізовано на серверній частині програмного застосунку у вигляді функції `loadUsersCollection()`.

```
async function loadUsersCollection() {
  const client = await
  mongodb.connect(      "mongodb+srv://Andrew:<password>@cluster0-
e77xk.mongodb.net/test?retryWrites=true",
    {
      useNewUrlParser: true
    }
  );
}
```

```

return client.db('test').collection('users');
}

```

4.3 Інтерфейс користувача

Кабінет користувача – це основне робоче місце користувача в системі. Він включає в себе інші підмодулі, які виконують основні функції системи, та є елементом компонування в системі. Доступ до системи користувача здійснюється шляхом авторизації через відповідну форму за допомогою Логіну та Пароля.

Головне меню

Система управління процесом утворення водно з відновлюваної целюлозовмісної сировини у біореакторі

Кабінет користувача Довідка Вийти

Форма для введення інформації про сировину

Вхідні дані

Сировина: Оберіть сировину

Вміст лігніну (%): Вміст лігніну

Вміст целюлози (%): Вміст целюлози

Вміст мінеральних солей (%): Вміст мінеральних солей

Кислотність (pH): Кислотність

Вміст сухої речовини (г/дм3): Вміст сухої речовини

Подрібненість (мм): Подрібненість

Маса сировини (кг): Маса сировини

Розрахувати

Кнопка для розрахунку

Результат

Результат розрахунку

ФОРМА НЕ ЗАПОВНЕНА

Повідомлення для користувача

Активация Windows
Переключитесь на версию "Настольная" или активируйте Windows

Рисунок 4.3 — сторінка “Кабінет користувача”

Підмодулі кабінету користувача:

До цих модулів належать наступні:

- модуль відображення кроків процесу;
- модуль відображення етапів процесу;
- модуль побудови графіка.

Модуль відображення кроків процесу та модуль відображення етапів процесу дозволяють відобразити результат роботи системи шляхом циклічного додавання елементів в DOM. Такий підхід відображення даних є гнучким, простим у використанні та дає можливість використовувати модулі повторно.

Модуль побудови графіка використовується для візуалізації даних про очікувану кількість водню в залежності від попередньої обробки целюлозовмісної сировини та її кількості.

Висновки до розділу

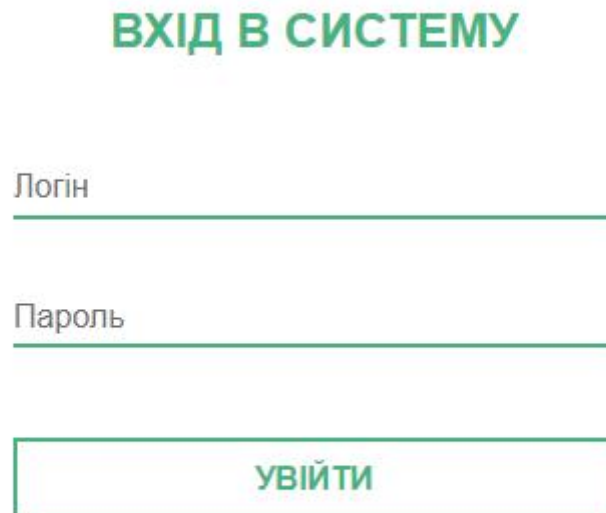
У даному розділі були розглянуті особливості реалізації програмного забезпечення. Окрім цього була описана реалізації бази даних. Також було описано інтерфейс користувача.

5. РОБОТА КОРИСТУВАЧА В СИСТЕМІ

Розроблена програмний комплекс розроблений з використанням веб-технологій і тому працює в браузерях, які підтримують актуальні веб-стандарти.

5.1 Форма авторизації користувача

При вході на клієнтський додаток, користувачеві необхідно авторизуватися в системі щоб почати працювати в системі. На рисунку 5.1 зображена форма авторизації.



The image shows a login form titled "ВХІД В СИСТЕМУ" (Login to the system) in green text. Below the title are two input fields: "Логін" (Login) and "Пароль" (Password), both with green borders. Below these fields is a green button labeled "УВІЙТИ" (Login).

Рисунок 5.1 — Форма авторизації

Зразок успішної авторизації показана на Рисунку 5.2

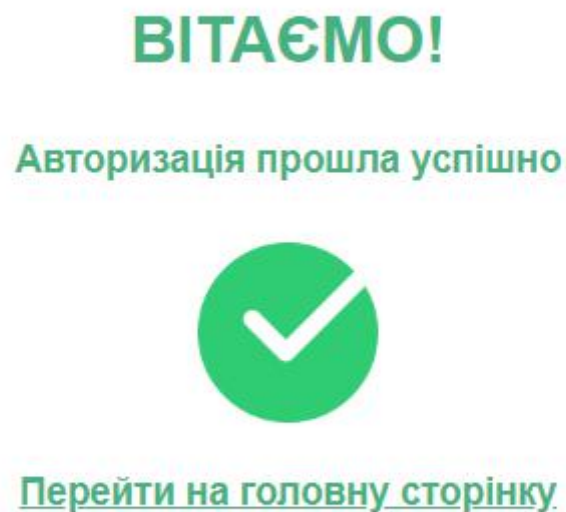


Рисунок 5.2 — Успішна авторизація користувача

5.2 Кабінет користувача

Після того, як користувач авторизувався в системі, він отримує доступ до головного меню системи. За допомогою цього меню користувач має доступ до усіх сторінок системи, або має можливість вийти з свого профілю. Також він може перейти до свого основного робочого простору – кабінету користувача. На рисунку 5.3 зображене головне меню системи.



Рисунок 5.3 — Головне меню системи

Перед роботою з системою користувач може переглянути сторінку Довідка.

На сторінці знаходиться основна інформація про призначення систему та методи роботи з системою. Також на сторінці є приклад роботи програмного застосунку.

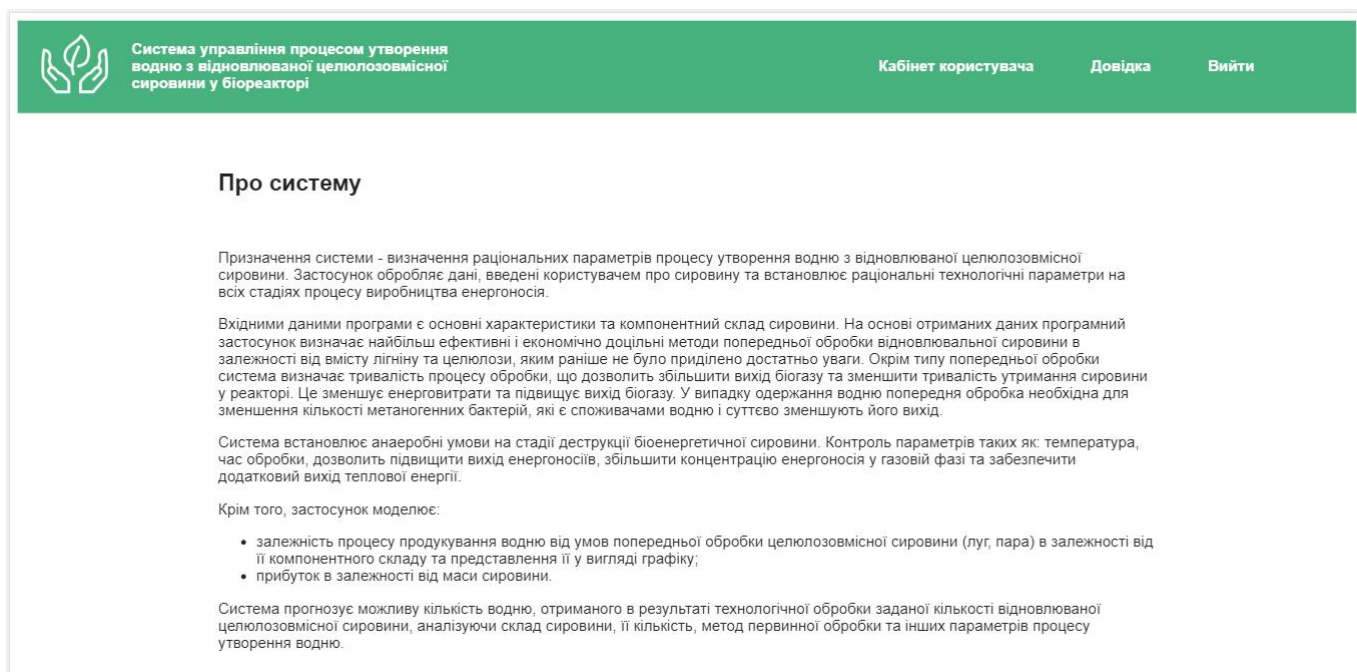


Рисунок 5.4 — Довідка

Одним із головних компонентів кабінету користувача є визначення параметрів процесу на основі вхідних даних. На рисунку 5.5 зображена для введення вхідних даних.

Вхідні дані

Сировина <input type="text" value="Обертіть сировину"/>	Вміст лігніну (%) <input type="text" value="Вміст лігніну"/>	Вміст целюлози (%) <input type="text" value="Вміст лігніну"/>
Вміст мінеральних солей (%) <input type="text" value="Вміст мінеральних солей"/>	Кислотність (pH) <input type="text" value="Кислотність"/>	Вміст сухої речовини (г/дм3) <input type="text" value="Вміст сухої речовини"/>
Подрібненість (мм) <input type="text" value="Подрібненість"/>	Маса сировини (кг) <input type="text" value="Маса сировини"/>	<input type="button" value="Розрахувати"/>

Рисунок 5.5 — Форма “Вхідні дані”

При некоректному заповненні форми, користувачу будуть показані підказки. Приклад повідомлення для користувача зображено на рисунку 5.6

Вхідні дані

Сировина <input type="text" value="солома ячменю"/>	Вміст лігніну (%) <input type="text" value="0% - 10%"/>	Вміст целюлози (%) <input type="text" value="20% - 30%"/>	Вміст мінеральних солей (%) <input type="text" value="45"/>
Кислотність (pH) <input type="text" value="25"/>	Вміст сухої речовини (г/дм3) <input type="text" value="45"/>	Подрібненість (мм) <input type="text" value="3-100 мм"/>	Маса сировини (кг) <input type="text" value="100"/>
<input type="button" value="Розрахувати"/>			

Результат

НЕ КОРЕКТНІ ВХІДНІ ДАНІ

ПЕРЕВІРТЕ ВХІДНІ ПОЛЯ:

Вміст мінеральних солей (%) (0 < 'Вміст мінеральних солей (%)' < 15)

Кислотність (pH) (0 < 'Кислотність (pH)' < 13)

Рисунок 5.6 — Повідомлення для користувача

Після заповнення форма користувач може отримати результат, натиснувши на кнопку “Розрахувати”. Якщо дані про сировину коректні, система відобразить результат в окремому вікні “Результат”. Приклад отриманого результату зображено на рисунку 5.7 та рисунку 5.8.

Результат

Основні параметри процесу

Попередня обробка сировини

Подрібнення біоенергетичної сировини	dч = 1-3 мм
Метод попередньої обробки	пар
Тривалість попередньої обробки	t = 1 год
Температура при попередній обробці	T = 130°C
Тиск при попередній обробці	P = 250 кПа

Рисунок 5.7 — Частина отриманого результату

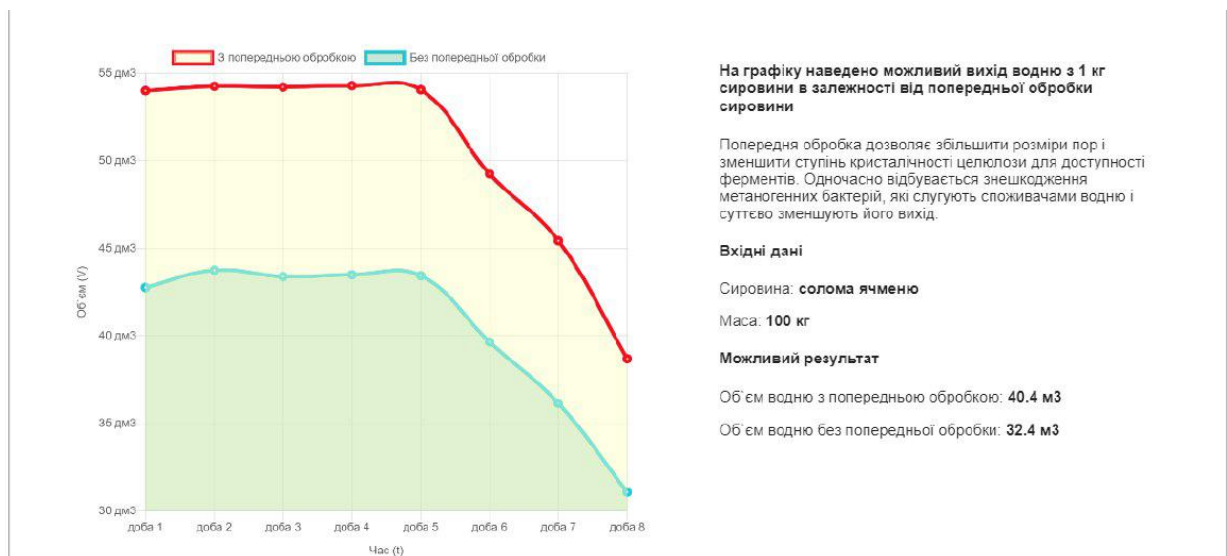


Рисунок 5.8 — Графік виходу водню

Висновки до розділу

У даному розділі був описаний інтерфейс користувача. Було описано основні сценарії роботи користувача з програмним продуктом. Показано приклад роботи системи.

ВИСНОВКИ

У ході виконання даної роботи було розроблено систему моделювання виходу водню від умов попередньої обробки сировини.

Створений програмний продукт дозволяє визначити технологічні параметри для процесу видобутку водню з відновлюваної целюлозовмісної сировини. Застосунок обробляє дані, введені користувачем про сировину та встановлює раціональні технологічні параметри на всіх стадіях процесу виробництва енергоносія. Вхідними даними програми є основні характеристики та компонентний склад сировини. На основі отриманих даних програмний застосунок визначає найбільш ефективні і економічно доцільні методи попередньої обробки сировини в залежності від вмісту лігніну та целюлози, яким раніше не було приділено достатньо уваги. Окрім типу попередньої обробки система визначає тривалість процесу обробки, що дозволить збільшити вихід біогазу та зменшити тривалість утримання сировини у реакторі. Це зменшує енерговитрати та підвищує вихід біогазу. У випадку одержання водню попередня обробка необхідна для зменшення кількості метаногенних бактерій, які є споживачами водню і суттєво зменшують його вихід.

Система встановлюватиме анаеробні умови на стадії деструкції біоенергетичної сировини. Контроль параметрів таких як: температура, час обробки, дозволить підвищити вихід енергоносіїв, збільшити концентрацію енергоносія у газовій фазі та забезпечити додатковий вихід теплової енергії.

За допомогою сучасних технологій було створено зручний користувацький інтерфейс.

Система спроектована із урахуванням потенційного розширення функціоналу у майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jeremy Ramsden. (2009). Bioinformatics. London: Springer Verlag P. 8-36.
2. Rick Riolo. (2016). Population Dynamics of Infectious Diseases: Theory and Applications. London: Chapman and Hall. P. 6-5
3. Markus Egger — MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF [Електронний ресурс]. — 2012. — Режим доступу: <https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf>.
4. Все про водень. [Електронний ресурс]. Режим доступу: http://kovtun-vm.at.ua/news/vse_pro_voden/2017-09-25-9
5. Голуб Г. А. Біогаз / Г. А. Голуб, В. О. Дубровін, В. М. Поліщук, К. М. Сєра, С. В. Драгнєв, О. А. Марус, О. В. Сидорчук, М. Ю. Павленко, В. В. Чуба, С. М. Кухарець // Проект «Підвищення енергоефективності та 189 стимулювання використання відновлювальної енергії а аграрно-харчових та інших малих та середніх підприємствах (МСП) України «Серія навчально-методичних матеріалів». – 2015. – Київ. – 48 с
6. Одержання біоводню в анаеробних процесах / Н.Б. Голуб, Д.І. Жураховська, К.В. Нікуліна, Н.В. Нікуліна // Відновлювальна енергетика, 2009. № 2. С. 65.
7. Rozendal, R.A.; Hamelers, H.V.M.; Euverink, G.J.W.; Metz, S.J.; Buisman, C.J.N. 2006. Principle and perspectives of hydrogen production through biocatalyzed electrolysis. International Journal of Hydrogen Energy. N 31: 1632 1640.
8. Кузык Б.Н. На пути к водородной энергетике / Б.Н. Кузык, В.И. Кушлин, Ю.В. Яковец. Москва: Ин-т эконом. стратегий. 2005. 160 с.
9. Цыганков А.А. Получение водовода биологическим путем / А.А. Цыганков // Российский химический журнал (Журнал Российского химического общества им. Д.И. Менделеева). 2006. Т. L. № 6. С. 33

- 10.El-Sakhawy, M.; Ha, M.L. 2007. Physical and mechanical properties of microcrystalline cellulose prepared from agricultural residues. Carbohydrate Polym. N 67: 1 10
- 11.Марков С.А. Биоводород: возможное использование водоростей и бактерий для получения молекулярного водорода / С.А. Марков // Альтернативная энергетика и экология. 2007. Т. 45, № 1. С. 30 35.
- 12.Ganesh, D. Saratale; Shing-Der, Chen; Yung-Chung, Lo; Rijuta, G. Saratale; Jo-Shu, Chang. 2008. Outlook of biohydrogen production from lignocellulosic feedstock using dark fermentation a review. Journal of Scientific and Industrial Research. Vol. 67: 962 979.
13. Lemi, Turker; Selcuk, Gumus; Alper, Tapan. 2008. Biohydrogen production: molecular aspects. Journal of Scientific and Industrial Research. Vol. 67: 994 1016.
14. Vue.js documentation [Электронный ресурс]. — Режим доступа: <https://vuejs.org/v2/guide/>
- 15.Martin Fowler — GUI Architectures. Часть 1 [Электронный ресурс]. — 2009. — Режим доступа: <https://bit.ly/2CvCk1e>.
16. Node.js documentation [Электронный ресурс]. — Режим доступа: <https://nodejs.org/uk/docs/>
- 17.Martin Fowler — GUI Architectures. Часть 2 [Электронный ресурс]. — 2009 — Режим доступа: <https://habr.com/post/53536/>.
18. MongoDB documentation [Электронный ресурс]. — Режим доступа: <https://docs.mongodb.com/>
- 19.Sass documentation [Электронный ресурс]. — Режим доступа: <https://sass-lang.com/>
20. Introducing JSON [Электронный ресурс]. — Режим доступа:<https://www.json.org/>

ДОДАТОК 1

Система моделювання залежності виходу водню від умов попередньої
обробки сировини.

Специфікація

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТІ51173_18Б

Аркушів 1

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТІ51173_1 8Б	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТІ51173_1 8Б 10-1	Main.vue	Основна компонента
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТІ51173_1 8Б 11-1	Додаток 3.doc	Опис головної компоненти

ДОДАТОК 2

Система моделювання залежності виходу водню від умов попередньої
обробки сировини.

Текст програми

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТІ51173_18Б 10-1

Аркушів 10

Київ 2019

```

<script>
import Multiselect from 'vue-multiselect';
import Steps from '../components/Step';
import Stages from '../components/Stage';
import Login from '../views/Login';
import ResultItem from '../components/ResultItem';
import Graph from '../components/Graph';
import Charts from '../components/Charts';
import jsPDF from 'jspdf';
import html2canvas from 'html2canvas';
export default {
  name: 'main',
  components: {
    Multiselect,
    Graph,
    Steps,
    Stages,
    Login,
    ResultItem,
    Charts,
  },
  data(){
    return{
      check: false,
      validationEmptyZero: true,
      validationZero: true,
      validationType: "",
      validationFormMessage: "",
      calcStatus: true,
      statusCheckFromIncorrectValues: true,
      statusCalcResultFirstPart: true,
      invalidFields: [],
      value1: null,
      valueLignin: null,
      valueSalts: null,
      valueDryMatter: '',
      valueShredding: null,
      valueWeight: null,
      valuePh: null,
      valueCellulose: null,
      selected: '',
      validationValue1: true,
      validationLignin: true,
      validationSalts: true,
      validationDryMatter: true,
      validationShredding: true,
      validationWeight: true,
      validationPh: true,
      validationCellulose: true,
      processDuration: 0,
      raw: ['солома пшениці', 'солома ячменю', 'відходи кукурудзи', 'відходи ріпаку',
'відходи соняшника', 'деревина берези', 'деревина сосни'],
      lignin: ['0% - 10%', '10% - 20%', '20% - 30%', '30% - 40%', '40% <'],

```

```

cellulose: ['0% - 10%', '10% - 20%', '20% - 30%', '30% - 40%', '40% <'],
shredding: ['1-3 мм', '3-100 мм', '100 мм <'],
fulness: ['> 30%', '30% - 40%', '40% - 50%', '50% - 60%', '60% - 70%', '70% - 80%', '80% - 90%', '90% - 100%'],
stages: [
  {id: 1, stageSet: 'stage1', stageName: 'Попередня підготовка сировини'},
  {id: 2, stageSet: 'stage2', stageName: 'Підготовка посівного матеріалу'},
  {id: 3, stageSet: 'stage3', stageName: 'Мікробіологічна деструкція біоенергетичної сировини'},
  {id: 4, stageSet: 'stage4', stageName: 'Відстоювання'},
  {id: 5, stageSet: 'stage5', stageName: 'Нейтралізація рідини для одержання водню'},
  {id: 6, stageSet: 'stage6', stageName: 'Одержання біоводню'},
  {id: 7, stageSet: 'stage7', stageName: 'Відстоювання'},
  {id: 8, stageSet: 'stage8', stageName: 'Очищення енергоносія (H2, CH4)'},
  {id: 9, stageSet: 'stage9', stageName: 'Одержання Na2CO3'}
],
allStages: {
  stage1: [],
  stage2: [],
  stage3: [],
  stage4: [],
  stage5: [],
  stage6: [],
  stage7: [],
  stage8: [],
  stage9: []
},
totalResult: {
  preProc: [],
  mainProc: [],
  labelsForChart: ''
},
base64Img: null,
}
},
watch: {
  valueDryMatter: function(val){
    if (val < 20){
      this.totalResult.labelsForChart = 264
    }
    if (val >= 20 && val < 29){
      this.totalResult.labelsForChart = 240
    }
    if (val >= 29 && val < 39){
      this.totalResult.labelsForChart = 216
    }
    if (val >= 39 && val < 49){
      this.totalResult.labelsForChart = 192
    }
    if (val >= 49 && val < 59){
      this.totalResult.labelsForChart = 168
    }
    if (val >= 59 && val < 69){
      this.totalResult.labelsForChart = 144
    }
  }
}

```

```

    }
    if (val >= 69){
        this.totalResult.labelsForChart = 120
    }
}
},
methods: {
    calcTotalResult: function (e) {
        this.check = false;
        this.check = true;
        this.totalResult.preProc.length = 0;
        this.totalResult.mainProc.length = 0;
        if ( this.checkFormEmpty() === true ){
            if (this.checkFromIncorrectValues() === true){
                this.calcResultFirstPart();
                this.calcResult();
            }
        }
    },
    checkFormEmpty: function (e) {
        this.validationValue1 = this.value1 !== null;
        this.validationLignin = this.valueLignin !== null;
        this.validationSalts = !(this.valueSalts === "" || this.valueSalts === null);
        this.validationDryMatter = !(this.valueDryMatter === "" || this.valueDryMatter ===
null);
        this.validationShredding = this.valueShredding !== null;
        this.validationWeight = !(this.valueWeight === "" || this.valueWeight === null);
        this.validationCellulose = this.valueCellulose !== null;
        this.validationPh = !(this.valuePh === "" || this.valuePh === null );
        if (this.validationValue1 === false ||
            this.validationLignin === false ||
            this.validationShredding === false ||
            this.validationSalts === false ||
            this.validationDryMatter === false ||
            this.validationWeight === false ||
            this.validationPh === false ||
            this.validationCellulose === false){
            this.validationEmptyZero = false;
            return false
        }
        else {
            this.validationEmptyZero = true;
            return true
        }
    },
    checkFromIncorrectValues: function(e){
        this.validationPh = true;
        this.validationSalts = true;
        this.validationDryMatter = true;
        this.validationWeight = true;
        this.statusCheckFromIncorrectValues = true;
        this.totalResult.preProc.length = 0;
        this.totalResult.mainProc.length = 0;
    }
}

```

```

this.invalidFields.length = 0;
if (this.valueSalts > 15 || this.valueSalts < 0 ){
    this.statusCheckFromIncorrectValues = false;
    this.invalidFields.push({text: "Вміст мінеральних солей (%) ( 0 < 'Вміст мінеральних солей (%)' < 15)"});
    this.validationSalts = false
}
if (this.valuePh > 13 || this.valuePh < 0){
    this.statusCheckFromIncorrectValues = false;
    this.invalidFields.push({text: "Кислотність (pH) ( 0 < 'Кислотність (pH)' < 13)" });
    this.validationPh = false
}
if (this.valueDryMatter > 150 || this.valueDryMatter <= 0){
    this.statusCheckFromIncorrectValues = false;
    this.invalidFields.push({text: "Вміст сухої речовини (г/дм3) ( 0 < 'Вміст сухої речовини (г/дм3)' < 150)"});
    this.validationDryMatter = false
}
if (this.valueWeight <= 0){
    this.statusCheckFromIncorrectValues = false;
    this.invalidFields.push({text: "Маса сировини (кг) ( 0 < 'Маса сировини (кг)' )"});
    this.validationWeight = false
}
return this.invalidFields.length === 0;
},
calcResultFirstPart: function(e){
    this.totalResult.preProc.length = 0;
    this.totalResult.mainProc.length = 0;
    this.validationType = '';
    this.statusCalcResultFirstPart = true;
    var inoculum = (this.valueWeight * 0.1667).toFixed(1);
    if (this.valueShredding !== '1-3 мм'){
        this.totalResult.preProc.push(
            {param: 'Подрібнення біоенергетичної сировини', value: 'dч = 1-3 мм'}
        )
    }
    if ( ((this.valueLignin === '0% - 10%' && this.valueCellulose === '30% - 40%') ||
        (this.valueLignin === '10% - 20%' && this.valueCellulose === '30% - 40%') ||
        (this.valueLignin === '0% - 10%' && this.valueCellulose === '40% <') ||
        (this.valueLignin === '10% - 20%' && this.valueCellulose === '40% <')) &&
        (this.value1 === 'солома ячменю' || this.value1 === 'відходи кукурудзи')){
        this.totalResult.preProc.push(
            {param: 'Метод попередньої обробки', value: 'пар'}
        );
        this.totalResult.preProc.push(
            {param: 'Тривалість попередньої обробки', value: 'τ = 1 год'}
        );
        this.totalResult.preProc.push(
            {param: 'Температура при попередній обробці', value: 'Т = 130°C'}
        );
        this.totalResult.preProc.push(
            {param: 'Тиск при попередній обробці', value: 'Р = 250 кПа'}
        );
        if (this.valuePh < 5){

```

```

        this.totalResult.preProc.push(
            {param: 'Кислотність при попередній обробці', value: 'pH = 5'}
        );
        this.totalResult.preProc.push(
            {param: 'Стабілізація кислотності', value: 'Оцтова кислота (CH3COOH)'}
        );
    }
    if (this.valueSalts < 5){
        this.totalResult.preProc.push(
            {param: 'Додавання мінеральних солей', value: 7 - this.valueSalts + ' %' }
        );
    }
    } else if ((this.valueLignin === '20% - 30%' || this.valueLignin === '30% - 40%' ||
this.valueLignin === '40% <') &&
        (this.valueCellulose === '0% - 10%' || this.valueCellulose === '10% - 20%'
|| this.valueCellulose === '20% - 30%')) &&
        (this.value1 === 'солома пшениці' || this.value1 === 'відходи ріпаку' ||
this.value1 === 'відходи соняшника' ||
        this.value1 === 'деревина берези' || this.value1 === 'деревина сосни'))){
        this.totalResult.preProc.push(
            {param: 'Метод попередньої обробки', value: 'луг (NaOH)'}
        );
        this.totalResult.preProc.push(
            {param: 'Концентрація луку', value: ' 2 моль/дм3'}
        );
        this.totalResult.preProc.push(
            {param: 'Тривалість попередньої обробки', value: ' τ = 2 год.'}
        );
        this.totalResult.preProc.push(
            {param: 'Температура при попередній обробці', value: 'Т = 120°C'}
        );
        this.totalResult.preProc.push(
            {param: 'Тиск при попередній обробці', value: 'Р = 180 кПа'}
        );
        if (this.valuePh < 5){
            this.totalResult.preProc.push(
                {param: 'Кислотність при попередній обробці', value: 'pH = 5'}
            );
            this.totalResult.preProc.push(
                {param: 'Стабілізація кислотності', value: 'Оцтова кислота (CH3COOH)'}
            );
        }
    }
    if (this.valueSalts < 5){
        this.totalResult.preProc.push(
            {param: 'Додавання мінеральних солей', value: 7 - this.valueSalts + ' %'}
        );
    }
    this.totalResult.preProc.push(
        {param: 'Перемішування при попередній обробці', value: 'постійне, 1 - 10
об/хв'}
    );
    this.totalResult.preProc.push(
        {param: 'Нейтралізація луку після попередньої обробки', value: 'стічною
водою'}
    );

```

```

    );
}
else {
    this.statusCalcResultFirstPart = false;
    this.validationType = "case1";
    return false
}

this.totalResult.mainProc.push(
    {param: 'Температура', value: 'Т = 35°C'}
);
this.totalResult.mainProc.push(
    {param: 'Концентрація сировини', value: '30 г/дм3'}
);
this.totalResult.mainProc.push(
    {param: 'Вміст інокуляту', value: inoculum.toString() + ' кг' }
);
this.totalResult.mainProc.push(
    {param: 'Кислотність', value: 'pH = 7'}
);
this.totalResult.mainProc.push(
    {param: 'Стабілізація кислотності', value: 'Оцтова кислота (CH3COOH)'}
);
this.totalResult.mainProc.push(
    {param: 'Перемішування', value: 'Періодичне, 10 об/хв'}
);
this.totalResult.mainProc.push(
    {param: 'Період перемішування', value: '5 год'}
);
this.totalResult.mainProc.push(
    {param: 'Тривалість перемішування', value: '10 хв'}
);
if (this.valueDryMatter < 20){
    this.totalResult.mainProc.push(
        {param: 'Тривалість процесу', value: 'τ = 264 год.'}
    );
}
if (this.valueDryMatter >= 20 && this.valueDryMatter < 29){
    this.totalResult.mainProc.push(
        {param: 'Тривалість процесу', value: 'τ = 240 год.'}
    );
}
if (this.valueDryMatter >= 29 && this.valueDryMatter < 39){
    this.totalResult.mainProc.push(
        {param: 'Тривалість процесу', value: 'τ = 216 год.'}
    );
}
if (this.valueDryMatter >= 39 && this.valueDryMatter < 49){
    this.totalResult.mainProc.push(
        {param: 'Тривалість процесу', value: 'τ = 192 год.'}
    );
}
if (this.valueDryMatter >= 49 && this.valueDryMatter < 59){
    this.totalResult.mainProc.push(

```

```

        {param: 'Тривалість процесу', value: 'τ = 168 год.'}
    );
}
if (this.valueDryMatter >= 59 && this.valueDryMatter < 69){
    this.totalResult.mainProc.push(
        {param: 'Тривалість процесу', value: 'τ = 144 год.'}
    );
}
if (this.valueDryMatter >= 69){
    this.totalResult.mainProc.push(
        {param: 'Тривалість процесу', value: 'τ = 120 год.'}
    );
}
},
calcResult: function(e) {
    this.allStages.stage1.length = 0;
    this.allStages.stage2.length = 0;
    this.allStages.stage3.length = 0;
    this.allStages.stage4.length = 0;
    this.allStages.stage5.length = 0;
    this.allStages.stage6.length = 0;
    this.allStages.stage7.length = 0;
    this.allStages.stage8.length = 0;
    this.allStages.stage9.length = 0;
    var stepCount = 0;
    if ((this.valueLignin === '20% - 30%' || this.valueLignin === '30% - 40%' ||
this.valueLignin === '40% <') &&
        (this.valueCellulose === '0% - 10%' || this.valueCellulose === '10% - 20%' ||
this.valueCellulose === '20% - 30%') &&
        (this.value1 === 'солома пшениці' || this.value1 === 'відходи ріпаку' ||
this.value1 === 'відходи соняшника' ||
        this.value1 === 'деревина берези' || this.value1 === 'деревина сосни')){
        this.allStages.stage1.push({
            stepID: stepCount + 1, stepDesc: 'Підготовка розчину лугу', stepParam: 'C=2
моль/дм3, насичений'
        });
        stepCount = stepCount + 1;
        if (this.valueShredding !== '1-3 мм'){
            this.allStages.stage1.push({
                stepID: stepCount + 1, stepDesc: 'Подрібнення біоенергетичної сировини',
stepParam: 'dч=1-3 мм'
            });
            this.allStages.stage1.push({
                stepID: stepCount + 2, stepDesc: 'Попередня обробка сировини', stepParam: 'pH =
5, P=180 кПа, T=120°C, τ=2 год.'
            });
            stepCount = stepCount + 2;
        }
        else {
            this.allStages.stage1.push({
                stepID: stepCount + 1, stepDesc: 'Попередня обробка сировини', stepParam: 'pH =
5, P=180 кПа, T=120°C, τ=2 год.'
            });
            stepCount = stepCount + 1;

```



```

    }
  }
  else if ((this.valueLignin === '0% - 10%' || this.valueLignin === '10% - 20%') &&
    (this.valueCellulose === '30% - 40%' || this.valueCellulose === '40% <') &&
    (this.value1 === 'солома ячменю' || this.value1 === 'відходи кукурудзи')){
    this.allStages.stage1.push({
      stepID: stepCount + 1, stepDesc: 'Підготовка пари', stepParam: 'T=120°C'
    });
    stepCount = stepCount + 1;
    if (this.valueShredding !== '1-3 мм'){
      this.allStages.stage1.push({
        stepID: stepCount + 1, stepDesc: 'Подрібнення біоенергетичної сировини',
stepParam: 'dч=1-3 мм'
      });
      this.allStages.stage1.push({
        stepID: stepCount + 2, stepDesc: 'Попередня обробка сировини', stepParam: 'Ph=5,
P=250 кПа, T=130°C, τ=1 год.'
      });
      stepCount = stepCount + 2;
    }
    else {
      this.allStages.stage1.push({
        stepID: stepCount + 1, stepDesc: 'Попередня обробка сировини', stepParam: 'Ph=5,
P=250 кПа, T=130°C, τ=1 год.'
      });
      stepCount = stepCount + 1;
    }
  }
  this.allStages.stage2.push({
    stepID: stepCount + 1, stepDesc: 'Отримання асоціації мікроорганізмів для деструкції
полімерних речовин ', stepParam: 'рН=7, T=35°C, n=10 об/хв.'
  });
  stepCount = stepCount + 1;
  this.allStages.stage2.push({
    stepID: stepCount + 1, stepDesc: 'Термічна обробка посівного матеріалу ', stepParam:
'T=90 °C, τ=30 хв.'
  });
  stepCount = stepCount + 1;
  this.allStages.stage2.push({
    stepID: stepCount + 1, stepDesc: 'Вирощування посівного матеріалу для одержання
водню ', stepParam: 'T=35°C, рН=7±0.5, n=10об/хв.'
  });
  stepCount = stepCount + 1;
  this.allStages.stage3.push({
    stepID: stepCount + 1, stepDesc: 'Мікробіологічна деструкція біоенергетичної
сировини', stepParam: 'T=35°C, рН=5±0.5, n=10об/хв.'
  });
  stepCount = stepCount + 1;
  this.allStages.stage4.push({
    stepID: stepCount + 1, stepDesc: 'Відстоювання', stepParam: 'τ=1 год, анаеробні
умови'
  });
  stepCount = stepCount + 1;
  this.allStages.stage5.push({

```

```

        stepID: stepCount + 1, stepDesc: 'Нейтралізація рідини для одержання водню',
stepParam: 'pH=7±0.5, n=10 об/хв, анаеробні умови'
    });
    stepCount = stepCount + 1;
    this.allStages.stage6.push({
        stepID: stepCount + 1, stepDesc: 'Одержання біоводню', stepParam: 'T=35 °C, pH=7±0.5,
n=10об/хв.'
    });
    stepCount = stepCount + 1;
    this.allStages.stage7.push({
        stepID: stepCount + 1, stepDesc: 'Відстоювання', stepParam: 'τ=9 год, анаеробні
умови'
    });
    stepCount = stepCount + 1;
    this.allStages.stage8.push({
        stepID: stepCount + 1, stepDesc: 'Очищення енергоносія від (CO2, H2S)', stepParam:
'v=0.5 м3/хв'
    });
    stepCount = stepCount + 1;
    this.allStages.stage8.push({
        stepID: stepCount + 1, stepDesc: 'Осушення CH4, або H2', stepParam: 'τ=2 год,
W=0,65'
    });
    stepCount = stepCount + 1;
    this.allStages.stage9.push({
        stepID: stepCount + 1, stepDesc: 'Одержання Na2CO3', stepParam: 'T=100 °C'
    });
    stepCount = stepCount + 1;
},
exportPDF() {
    var quotes = document.getElementById('export-container');
    var canvasElement = document.createElement('canvas');
    html2canvas(document.body, { canvas: canvasElement
    }).then(function (canvas) {
        var contentWidth = canvas.width;
        var contentHeight = canvas.height;
        //The height of the canvas which one pdf page can show;
        var pageHeight = contentWidth / 592.28 * 841.89;
        //the height of canvas that haven't render to pdf
        var leftHeight = contentHeight;
        //addImage y-axial offset
        var position = -48;
        //a4 format [595.28,841.89]
        var imgWidth = 595.28;
        var imgHeight = 592.28/contentWidth * contentHeight;
        var pageData = canvas.toDataURL('image/jpeg', 1.0);
        var pdf = new jsPDF('', 'pt', 'a4');
        if (leftHeight < pageHeight) {
            pdf.addImage(pageData, 'JPEG', 0, 0, imgWidth, imgHeight );
        } else {
            while(leftHeight > 0) {
                pdf.addImage(pageData, 'JPEG', 0, position, imgWidth, imgHeight)
                leftHeight -= pageHeight;
                position -= 841.89;
            }
        }
    });
}

```

```

        //avoid blank page
        if(leftHeight > 0) {
            pdf.addPage();
        }
    }
    var obj = new Date();
    pdf.save('DMY-' + obj.getUTCDate() + '/' + (obj.getUTCMonth() + 1) + '/' +
            obj.getUTCFullYear() + '-' + obj.getTime() + '.pdf');
    pdf.output('dataurlnewwindow');
    });
},
}
</script>

```

ДОДАТОК 3

Система моделювання залежності виходу водню від умов попередньої
обробки сировини.

Опис програми

УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТІ51173_18Б 11-1

Аркушів 8

Київ 2019

АНОТАЦІЇ

Розділ містить опис частини, яка слугує для визначення методу попередньої обробки на основі введених користувачем даних, встановлення технологічних параметрів стадії попередньої обробки сировини та встановлення технологічних параметрів для основної стадії процесу видобутку водню.

Описаний модуль є реалізацію системи системи моделювання залежності виходу водню від умов попередньої обробки сировини.

Модуль написано мовою програмування JavaScript.

ЗМІСТ

1. Загальні відомості	71
2. Функціональне призначення	72
3. Опис логічної структури	73
4. Технічні засоби, що використовуються	74
5. Виклик і завантаження	75
6. Вхідні та вихідні дані	76

ЗАГАЛЬНІ ВІДОМОСТІ

У додатку розглядається один з програмних модулів системи — модуль для визначення методу попередньої обробки та основних параметрів процесу видобутку водню з кодом УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_TI51173_18Б 10-1, що міститься у файлі Main.vue. Модуль реалізовано за допомогою фреймворка Vue.js. Модуль призначений для визначення методу попередньої обробки на основі введених користувачем даних, встановлення технологічних параметрів стадії попередньої обробки сировини та встановлення технологічних параметрів для основної стадії процесу видобутку водню.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Призначенням модулю є визначення методу попередньої обробки на основі введених користувачем даних, встановлення технологічних параметрів стадії попередньої обробки сировини та встановлення технологічних параметрів для основної стадії процесу видобутку водню. Компонента Main являє собою кабінет користувача в якому відбуваються всі розрахунки. Використання такого шаблону дозволяє створювати програмне забезпечення, де результат роботи системи буде відображатися на HTML-сторінці без її перезавантаження.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Слідкувати за зміною вхідних даних та розрахунок результату на основі введених користувачем даних є головним завданням модуля. Модуль обробляє інформацію, перевіряє введені користувачем дані, надає користувачеві підказки щодо коректності введених даних про сировину, проводить розрахунки та виводить результат роботи на HTML-сторінку без її перезавантаження.

ТЕХНІЧНІ ЗАСОБИ ЩО ВИКОРИСТОВУЮТЬСЯ

Модуль розроблено у середовищі розробки JetBrains WebStorm 2018, що забезпечує набір сервісних функцій та графічний діалог з користувачем, на комп'ютері, що використовував операційну систему Windows 10.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Програмний модуль реалізований як vue-компонент, який забезпечує існування клієнтської частини та бізнес-логіки разом. Для використання даного модулю не потрібно ніяких дій, оскільки він автоматично спрацьовує після запуску клієнтського веб-додатку.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для модуля є інформація про компонентний склад сировини, яку користувач вводить у відповідну форму на сторінці “Кабінет користувача”.

Вихідними даними програмного модуля є технологічні параметри стадії попередньої обробки сировини та технологічні параметри основної стадії процесу видобутку водню.

ДОДАТОК 4

Інформаційна система ефективного менеджменту роботи закладів
післядипломної освіти

Апробації

УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТІ51173_18Б

Аркушів 3

Київ 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної
науково-практичної конференції
молодих вчених та студентів
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ - 2019

Природні ресурси регіону. Система обліку лісових ресурсів.	164
<i>ЛІНЬОВ Д.О., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
Інтернет-система редагування та аналізу генетичних послідовностей.	165
<i>ЖИРОВ М.І., студент гр. ТІ-51</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
Google maps API. Обробка даних отриманих з пристрою та їх відображення на карті веб-інтерфейсу.	166
<i>ЖИРНОВ А.Ю., студент гр. ТВ-51</i>	
<i>Керівник - доц., к.т.н. Смаковський Д.С.</i>	
Природні ресурси регіону. Система обліку водних ресурсів.	167
<i>ДАВИДЧУК О.С., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
Інформаційна система біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій.	168
<i>ГОРБАТЮК О.І., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення та прогнозу статусу гравця/команди гравців.	169
<i>ГОНЧАР О.В., студент гр. ТР-52</i>	
<i>Керівник - доц., к.т.н. Коваль О.В.</i>	
Природні ресурси регіону. Система обліку лікувальних ресурсів.	170
<i>ГАРКУША О.В., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
Система управління процесом утворення водню з відновлюваної целюлозовмісної сировини у біореакторі.	171
<i>БОНДАРЕНКО А.В., студент гр. ТІ-51</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
Автоматизована система для моделювання та аналізу адміністративних функцій організацій.	172
<i>БИТИК М.О., студент гр. ТМ-51</i>	
<i>Керівник - доц., к.т.н. Кузьмініх В.О.</i>	
Інформаційна система ефективного менеджменту роботи закладів післядипломної освіти.	173
<i>БАСАЛИК Д.А., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
Інформаційна система аналізу профілактики хронічних вірусних інфекцій.	174
<i>БАСАЛИК Г.А., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
Розробка сайту для продажу продуктів здорового харчування.	175
<i>СЕМІОН В.І., студент гр. ТМ-62</i>	
<i>Керівник - ст.викл., Шульженко О.Ф.</i>	
Організація по впровадженню та обслуговуванню автоматизованого обладнання теплиць.	176
<i>ЛОКОТАРЬОВ Є.О., студент гр. ТМ-62;</i>	
<i>ШАПОВАЛ В.О., студент гр. ТМ-62</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
Розробка прикладного програмного забезпечення для автоматизації роботи кінологічного клубу.	177
<i>ЛЕБЕДИК Т.О., студент гр. ТМ-62;</i>	
<i>САПОН О.М., студент гр. ТМ-62</i>	

СИСТЕМА УПРАВЛІННЯ ПРОЦЕСОМ УТВОРЕННЯ ВОДНЮ З ВІДНОВЛЮВАНОЇ ЦЕЛЮЛОЗОВІСНОЇ СИРОВИНИ У БІОРЕАКТОРІ

Сучасні технології одержання водню є енергоємними та економічно не раціональними. Основні проблеми виникають на етапі попередньої обробки целюлозовмісної сировини, оскільки існуючі технології не враховують особливостей компонентного складу сировини.

Метою роботи є створення програмного продукту, результатом роботи якого буде визначення раціональних параметрів процесу утворення водню з відновлюваної целюлозовмісної сировини[1]. Застосунок має обробляти дані, введені користувачем про сировину та встановлювати раціональні технологічні параметри на всіх стадіях процесу виробництва енергоносія.

Вхідними даними програми є основні характеристики та компонентний склад сировини. На основі отриманих даних програмний застосунок визначатиме найбільш ефективні і економічно доцільні методи попередньої обробки відновлювальної сировини в залежності від вмісту лігніну та целюлози, яким раніше не було приділено достатньо уваги. Окрім типу попередньої обробки система має визначати тривалість процесу обробки, що дозволить збільшити вихід біогазу та зменшити тривалість утримання сировини у реакторі. Це зменшує енерговитрати та підвищує вихід біогазу. У випадку одержання водню попередня обробка необхідна для зменшення кількості метаногенних бактерій[2], які є споживачами водню і суттєво зменшують його вихід.

Система встановлюватиме анаеробні умови на стадії деструкції біоенергетичної сировини. Контроль параметрів таких як: температура, час обробки, дозволить підвищити вихід енергоносіїв, збільшити концентрацію енергоносія у газовій фазі та забезпечити додатковий вихід теплової енергії.

Крім того, застосунок має моделювати:

- залежність процесу продукування водню від умов попередньої обробки целюлозовмісної сировини (кислота, луг, пара, ультразвук) в залежності від її компонентного складу та представлення її у вигляді графіку;
- залежність виходу водню від умов попередньої обробки інокуляту (пара, струм) та представлення її у вигляді графіку.
- процес утворення водню в залежності від співвідношення інокулят/целюлозовмісна сировина різного складу.

Система прогнозуватиме можливу кількість водню, отриманого в результаті технологічної обробки заданої кількості відновлюваної целюлозовмісної сировини, аналізуючи склад сировини, її кількість, метод первинної обробки та інших параметрів процесу утворення водню.

Програмний застосунок забезпечить раціональне використання економічних та природних ресурсів, прискорить процес виробництва та збільшить обсяги готового енергетичного продукту.

Перелік посилань:

1. Jeremy Ramsden. (2009). Bioinformatics. London: Springer Verlag P. 8-36..
2. Rick Riolo. (2016). Population Dynamics of Infectious Diseases: Theory and Applications. London: Chapman and Hall. P. 6-58.